

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**  
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**ПО ДИСЦИПЛИНЕ**

**«Web-программирование»**

**НА ТЕМУ:**

**«Работа с сокетами»**

**Работу выполнила:**

**Студентка гр. К33422**

**Орлова Любовь Дмитриевна**

**Преподаватель:**

**Говоров Антон Игоревич**

**Санкт-Петербург**

**2020**

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

Практическое 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.

server.py

```
1 import socket
2
3 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4
5 sock.bind(('', 9090))
6 sock.listen(1)
7 conn, addr = sock.accept()
8 print('connected:', addr)
9
10 while True:
11
12     data = conn.recv(1024)
13     print(data.decode("utf-8"))
14     if not data:
15         break
16     conn.send(bytes("Hello, client!", "utf-8"))
17
18 conn.close()
```

client.py

```
1 import socket
2
3 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 sock.connect(('localhost', 9090))
5 sock.send(bytes("Hello, server!", "utf-8"))
6
7 msg = sock.recv(1024)
```

```
8 print(msg.decode("utf-8"))
9
10 sock.close()
```

Выполнение:

```
C:\Users\orl\Web\part1>python server.py
connected: ('127.0.0.1', 49631)
Hello, server!
```

```
C:\Users\orl\Web\part1>python client.py
Hello, client!
```

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 b: Решение квадратного уравнения.

server.py

```
1 import socket
2 import math
3
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 sock.bind(('', 9090))
7 sock.listen(1)
8 conn, addr = sock.accept()
9 print('connected:', addr)
10
11 conn.send(bytes(" Введите a: ", "utf-8"))
12 a = conn.recv(1024)
13 print(a.decode("utf-8"))
14
15 conn.send(bytes(" Введите b: ", "utf-8"))
16 b = conn.recv(1024)
17 print(b.decode("utf-8"))
18
19 conn.send(bytes(" Введите c: ", "utf-8"))
20 c = conn.recv(1024)
21 print(c.decode("utf-8"))
```

```

22
23 a = int(a)
24 b = int(b)
25 c = int(c)
26
27 D = b * b - 4 * a * c
28
29 if D < 0 :
30     conn.send(bytes("Корней нет(D < 0)", "utf-8"))
31 else:
32     x1 = str((- b + math.sqrt(D)) / ( 2 * a ))
33     x2 = str((- b - math.sqrt(D)) / ( 2 * a ))
34     print(x1)
35     print(x2)
36     conn.send(bytes("Ответ: x1 = %s, x2 = %s" % (x1, x2), "utf-8"))
37
38 conn.close()

```

### client.py

```

1 import socket
2
3 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 sock.connect(('localhost', 9090))
5
6 a = sock.recv(1024)
7 print(a.decode("utf-8"))
8 num = input()
9 sock.send(bytes(num, "utf-8"))
10
11 b = sock.recv(1024)
12 print(b.decode("utf-8"))
13 num = input()
14 sock.send(bytes(num, "utf-8"))
15
16 c = sock.recv(1024)
17 print(c.decode("utf-8"))
18 num = input()
19 sock.send(bytes(num, "utf-8"))
20
21 msg = sock.recv(1024)
22 print(msg.decode("utf-8"))
23

```

```
24 sock.close()
```

Выполнение:

```
C:\Users\orl\Web\part2>server.py
connected: ('127.0.0.1', 49679)
1
-1
-6
3.0
-2.0
```

```
C:\Users\orl\Web\part2>client.py
Введите a:
1
Введите b:
-1
Введите c:
-6
Ответ: x1 = 3.0, x2 = -2.0
```

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.

server.py

```
1 import socket
2 import time
3
4 server = socket.socket()
5 host = 'localhost'
6 port = 555
7 server.bind((host,port))
8
9 print ('Starting server on', host, port)
10
11 server.listen(5)
12
13 while True:
14     try:
15
16         client, (client_host, client_port) = server.accept()
17         print('New connection', client_host, client_port)
18         response_type = 'HTTP/1.0 200 OK\n'
```

```

19         headers = 'Content-Type: text/html\n\n'
20         content = open('index.html', 'r')
21         body = ''.join(content)
22         response = response_type + headers + body
23         client.send(response.encode('utf-8'))
24         client.close()
25     except KeyboardInterrupt:
26         server.close()
27         break

```

### client.py

```

1 import socket
2
3 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 sock.connect(('localhost', 555))
5 msg = sock.recv(1024)
6 print(msg.decode("utf-8"))
7 sock.close()

```

Выполнение:

```

C:\Users\orl\Web\part3>python server.py
Starting server on localhost 555
New connection 127.0.0.1 62114

```

```

C:\Users\orl\Web\part3>python client.py
HTTP/1.0 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Hello world!</title>

  <!-- CSS -->
  <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
  <style>
    body { padding-top:50px; }
  </style>
</head>
<body class="container">

  <div class="jumbotron">
    <h1>Hello world!</h1>
  </div>
</body>
</html>

```

4. Реализовать многопользовательский чат.

### server.py

```

1 import threading
2 import socket
3
4 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 server.bind(('127.0.0.1', 5555))
6 server.listen()
7
8 clients = []
9 names = []
10 addressss = []
11
12 def broadcast(message):
13     for client in clients:
14         client.send(message)
15
16 def chat(client):
17     while True:
18         try:
19             message = client.recv(1024)
20             broadcast(message)
21         except:
22             index = clients.index(client)
23             clients.remove(client)
24             client.close()
25             name = names[index]
26             address = addressss[index]
27             broadcast(f'{name} left the chat!'.encode('ascii'))
28             print(f'Disconnected {address}')
29             names.remove(name)
30             addressss.remove(address)
31             break
32
33 def new_connection():
34     while True:
35         client, address = server.accept()
36         print(f"Connected with {str(address)}")
37
38         client.send('new_conn'.encode('ascii'))
39         name = client.recv(1024).decode('ascii')
40
41         online = ""
42         if len(names) == 0:

```

```

43             online = "You are the only person in the chat"
44         elif len(names) == 1:
45             online = names[0]
46             online += " is online!"
47
48         else:
49             for i, person in enumerate(names):
50                 online += f" {person}"
51                 if i != len(names) - 1:
52                     online += ","
53
54             online += " are online!"
55
56
57         names.append(name)
58         clients.append(client)
59         addressss.append(address)
60
61         broadcast(f'{name} joined the chat!'.encode('ascii'))
62         greeting = "Hi there! " + online
63         client.send(greeting.encode('ascii'))
64
65         thread = threading.Thread(target=chat, args=(client,))
66         thread.start()
67
68     print("Server is listening...")
69     new_connection()

```

### client.py

```

1  import socket
2  import threading
3  import time
4
5
6  name = input("What's your name? ")
7
8  client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9  client.connect(('127.0.0.1', 5555))
10
11 def receive():
12     while True:
13         try:

```



```

14         message = client.recv(1024).decode('ascii')
15         if message == 'new_conn':
16             client.send(name.encode('ascii'))
17         else:
18             itsatime = time.strftime("%Y/%m/%d - %H:%M",
time.localtime())
19             print("[",itsatime,"] ", message)
20         except:
21             print("Server has stopped working!")
22             client.close()
23             break
24
25 def write():
26     while True:
27         message = f'{name}: {input("")}'
28         client.send(message.encode('ascii'))
29
30 receive_thread = threading.Thread(target=receive)
31 receive_thread.start()
32
33 write_thread = threading.Thread(target=write)
34 write_thread.start()

```

## Выполнение:

```

C:\Users\orl\Web\part4>python client.py
What's your name? Nadya
[ 2020/09/24 - 13:06 ] Nadya joined the chat!
[ 2020/09/24 - 13:06 ] Hi there! Nastya is online!
[ 2020/09/24 - 13:06 ] Kate joined the chat!
[ 2020/09/24 - 13:07 ] Nastya: Hey what's up!
Hello!
[ 2020/09/24 - 13:07 ] Nadya: Hello!
[ 2020/09/24 - 13:07 ] Kate: Hi! :)

```

```

C:\Users\orl\Web\part4>python client.py
What's your name? Kate
[ 2020/09/24 - 13:06 ] Kate joined the chat!
[ 2020/09/24 - 13:06 ] Hi there! Nastya, Nadya are online!
[ 2020/09/24 - 13:07 ] Nastya: Hey what's up!
[ 2020/09/24 - 13:07 ] Nadya: Hello!
Hi! :)
[ 2020/09/24 - 13:07 ] Kate: Hi! :)
[ 2020/09/24 - 13:08 ] Nadya left the chat!

```

```
C:\Users\orl\Web\part4>python client.py
What's your name? Nastya
[ 2020/09/24 - 13:05 ] Nastya joined the chat!Hi there! You are the only person in the chat
[ 2020/09/24 - 13:06 ] Nadya joined the chat!
[ 2020/09/24 - 13:06 ] Kate joined the chat!
Hey what's up!
[ 2020/09/24 - 13:07 ] Nastya: Hey what's up!
[ 2020/09/24 - 13:07 ] Nadya: Hello!
[ 2020/09/24 - 13:07 ] Kate: Hi! :)
[ 2020/09/24 - 13:08 ] Nadya left the chat!
[ 2020/09/24 - 13:08 ] Kate left the chat!
```

```
C:\Users\orl\Web\part4>python server.py
Server is listening...
Connected with ('127.0.0.1', 64003)
Connected with ('127.0.0.1', 64006)
Connected with ('127.0.0.1', 64007)
Disconnected ('127.0.0.1', 64006)
Disconnected ('127.0.0.1', 64007)
Disconnected ('127.0.0.1', 64003)
```

Вывод: в ходе выполнения лабораторной работы были получены навыки реализации web-серверов с использованием сокетов и потоков на языке Python.