

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Веб-программирование

Отчет

Лабораторная работа №1
“Работа с сокетами”

Выполнила:
Каратецкая Мария

Группа К33402

Проверила:
Говоров Антон Игоревич

Санкт-Петербург

2021 г.

Цель работы

Овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.

Задание 1:

Client.py

```
import socket
import time
conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 1234))      #открытие сокета
conn.send(b"Hello, server! \n")      #отправка приветствия серверу
print('message send \n')
time.sleep(3)
data = conn.recv(22222)
udata = data.decode("utf-8")
print(udata)
conn.close
```

Server.py

```
import socket
import time
conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 1234))
conn.listen(10)

clientsocket, address = conn.accept()
data = clientsocket.recv(22222)
if data:
    udata = data.decode("utf-8")
    print(udata)
    time.sleep(3)
    clientsocket.sendall(b"Hello, client! \n")
    print('message send')
conn.close()
```

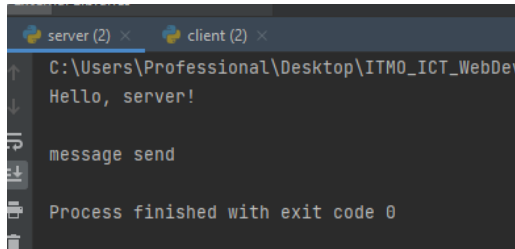
Приветствие клиента

```
server (2) x client (2) x
C:\Users\Professional\Desktop\ITMO_ICT_WebDevelopment_2021-2022
message send

Hello, client!

Process finished with exit code 0
```

Приветствие сервера



Задание 2:

Client.py

```
import socket
import time

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 1234))

a = str.encode(input('введите название желаемой функции \n'
    'Треугольник Пифагора - 1\n'
    'Квадратное уравнение - 2\n'
    'Площадь трапеции - 3\n'
    'Площадь параллелограмма - 4\n'))
conn.send(a)

while True:
    data = conn.recv(22222)
    udata = data.decode("utf-8")
    print(udata)
    answer = str.encode(input())
    conn.send(answer)

conn.close
```

Server.py

```
import socket
import time
import math

def pif(a,b):
    return str(math.sqrt(a*a + b*b))

def equation(a,b,c):
    d = b*b - 4*a*c
    if d == 0:
        return 'корень ' + str((math.sqrt(d) - b)/2/a)
    elif d > 0:
        x1 = (math.sqrt(d) - b)/2/a
        x2 = -1*(math.sqrt(d) + b)/2/a
        return str('1 корень ' + str(x1) + '\n2 корень ' + str(x2))
    else:
        return 'нет корней'

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 1234))
conn.listen(1)

clientsocket, address = conn.accept()
data = clientsocket.recv(22222)
```

```

if data:
    udata = data.decode("utf-8")
    if udata == '1':
        clientsocket.send(str.encode("Введите первый катет \n"))
        data = clientsocket.recv(22222)
        a = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите второй катет \n"))
        data = clientsocket.recv(22222)
        b = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode(pif(a,b)))

    if udata == '2':
        clientsocket.send(str.encode("Введите первый член квадратного уравнения \n"))
        data = clientsocket.recv(22222)
        a = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите второй член квадратного уравнения \n"))
        data = clientsocket.recv(22222)
        b = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите третий член квадратного уравнения \n"))
        data = clientsocket.recv(22222)
        c = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode(equation(a,b, c)))

```

```

    if udata == '3':
        clientsocket.send(str.encode("Введите первую сторону трапеции \n"))
        data = clientsocket.recv(22222)
        a = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите вторую сторону трапеции \n"))
        data = clientsocket.recv(22222)
        b = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите высоту трапеции \n"))
        data = clientsocket.recv(22222)
        c = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode(str((a+b)*c/2)))

    if udata == '4':
        clientsocket.send(str.encode("Введите сторону параллелограмма \n"))
        data = clientsocket.recv(22222)
        a = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode("Введите высоту параллелограмма \n"))
        data = clientsocket.recv(22222)
        b = int(data.decode("utf-8"))
        clientsocket.sendall(str.encode(str(a*b)))

conn.close()

```

Выполнение программы

```

07:03:51.9 (1707033101) > (вектор {1707_107_новбеск...
введите название желаемой функции
Треугольник Пифагора - 1
Квадратное уравнение - 2
Площадь трапеции - 3
Площадь параллелограмма - 4
1
Введите первый катет
2
Введите второй катет
3
6.324555320336759

```

Задание 3:

Server.py

```
import socket
import time

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("localhost", 8080))
conn.listen(1)

clientsocket, address = conn.accept()
data = clientsocket.recv(22222)
count = (data.decode("utf-8")).split(' ')

with open("index.html", "w") as file:
    file.write("<!DOCTYPE html> \n<html> \n<body> \n")

    for i in range(0, len(count) - 1, 2):
        file.write("<h2>" + 'Дисциплина: ' + count[i] + ' Баллы: ' + count[i+1] + "</h2>\n")
    file.write("\n</html> \n</body> \n")

page = open('index.html')
answer = page.read()
page.close()
response_type = 'HTTP/1.0 200 OK\n'
headers = 'Content-Type: text/html\n'
response = response_type + headers + answer
data = response.encode("utf-8")

clientsocket.sendall(data)
time.sleep(15)
conn.close()
```

Client.py

```
import socket
import time

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("localhost", 8080))
a = int(input('Введите количество дисциплин\n'))
list = ''
for i in range(a):
    list += (input('Введите ' + str(i+1) + ' дисциплину и количество баллов \n')) + ' '
conn.send(list.encode('utf-8'))
data = conn.recv(22222)
udata = data.decode()
print(udata)
data = conn.recv(22222)
udata = data.decode()
print(udata)
time.sleep(15)
conn.close()
```

Выполнение программы

```

C:\Users\Professional\Desktop\ITMO_ICT_WebDevelopment_2021-2022
Введите количество дисциплин
4
Введите 1 дисциплину и количество баллов
Веб-программирование 45
Введите 2 дисциплину и количество баллов
Операционные системы 30
Введите 3 дисциплину и количество баллов
Адаптивный веб-дизайн 55
Введите 4 дисциплину и количество баллов
Английский язык 30
HTTP/1.0 200 OK
Content-Type: text/html
<!DOCTYPE html>
<html>
<body>
<h2>Дисциплина: Веб-программирование Баллы: 45</h2>
<h2>Дисциплина: Операционные Баллы: системы</h2>
<h2>Дисциплина: 30 Баллы: Адаптивный</h2>
<h2>Дисциплина: веб-дизайн Баллы: 55</h2>
<h2>Дисциплина: английский Баллы: язык</h2>
<h2>Дисциплина: 30 Баллы: </h2>

</html>
</body>

```

Задание 4:

Server.py

```

import socket, threading

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = "127.0.0.1"
port = 9090
server.bind((host, port))

server.listen(5)

clients = list()
end = list()

def get():
    while True:
        client, addr = server.accept()
        clients.append(client)
        print(f'Сервер подключен через {addr}: количество клиентов: {len (clients)}', end = '\n')

def recv_data(client):
    while True:
        try:
            indata = client.recv(1024)
        except Exception:
            clients.remove(client)
            end.remove(client)
            print(f'Сервер отключен: количество клиентов: {len (clients)}', end = '\n')
            break
        print(indata.decode('utf-8'))
        for i in clients:
            if i != client:
                i.send(indata)

```

```

def send_mes():
    while True:
        print('')
        outdata = input('')
        print()
        for client in clients:
            client.send(f"Сервер: {outdata}".encode('utf-8'))

def get_mes():
    while True:
        for i in clients:
            if i in end:
                continue
            index = threading.Thread(target=recv_data, args=(i,))
            index.start()
            end.append(i)

t1 = threading.Thread(target=send_mes, name='input')
t1.start()
t2 = threading.Thread(target=get_mes, name='out')
t2.start()
t3 = threading.Thread(target=get(), name='get')
t3.start()
t2.join()

for i in clients:
    i.close()

```

Client{1..}.py

```

import socket
import threading

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = '127.0.0.1'

name = input('Введите свой ник\n')
port = 9090
client.connect((host, port))
print(name + ' ' + 'подключился к серверу\n')

def send_mes():
    while True:
        outdata = input('')
        print()
        client.send(f'{name}:{outdata}'.encode('utf-8'))
        print('%s:%s' % (name, outdata))

def get_mes():
    while True:
        indata = client.recv(1024)
        print(indata.decode('utf-8'))

t1 = threading.Thread(target=get_mes, name='input')
t2 = threading.Thread(target=send_mes, name='out')
t1.start()
t2.start()
t2.join()
print('сервер отключен')
client.close()

```

Server выполнение программы

```
client2 x client (1) x server (2) x client1 x
C:\Users\Professional\Desktop\ITMO_ICT_WebDevelopment_2021-2022\students\K3

сервер подключен через ('127.0.0.1', 55651): количество клиентов: 1
client1:ghjk
сервер подключен через ('127.0.0.1', 57205): количество клиентов: 2
client2:hjkl
client1:jkl\
client1:jkhk
```

Client1 выполнение программы

```
C:\Users\Professional\Desktop\ITMO_ICT_WebDevelopment_2021-2022\students\K3
Введите свой ник
client1
client1 подключился к серверу

ghjk

client1:ghjk
client2:hjkl
jkl\

client1:jkl\
jkhk

client1:jkhk
|
```

Client2 выполнение программы

```
C:\Users\Professional\Desktop\ITMO_ICT_WebDevelopment_2021-2022\students\K3
Введите свой ник
client2
client2 подключился к серверу

hjkl

client2:hjkl
client1:jkl\
client1:jkhk
|
```

Вывод:

В ходе данной лабораторной, я овладела практическими навыками и умениями реализации web-серверов и использования сокетов.