

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»
Направление подготовки «45.04.04 Интеллектуальные системы в гуманитарной среде»

О Т Ч Е Т
о выполнении лабораторной работы №1
«Работа с сокетом»

Выполнил:
студент группы № К33421
Змиевский Д. А.

Санкт-Петербург
2020

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

Оборудование: компьютерный класс.

Программное обеспечение: Python 2.7-3.6, библиотеки Python: sys, socket

Задание №1

Текст задания:

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.

Код:

```
server.py
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 9090))
conn.listen(10)

while True:
    try:
        clientsocket, address = conn.accept()
        data = clientsocket.recv(1024)
        udata = data.decode("utf-8")
        print(udata)
        clientsocket.send(b"Hello, client")
        clientsocket.close()
    except KeyboardInterrupt:
        conn.close()
        break
```

```
client.py
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 9090))
conn.send(b"Hello, server")

data = conn.recv(1024)
udata = data.decode("utf-8")
conn.close()

print(udata)
```

Результат:

```
(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task01$ python server.py
Hello, server

(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task01$ python client.py
Hello, client
```

Задание №2

Текст задания:

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. (Номер в ИСУ: 4; с. Поиск площади параллелограмма.)

Код:

```
server.py
import socket

def parall_area(data):
    a, h = map(int, data.split())
    S = a * h
    return str(S)

if __name__ == '__main__':
    conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    conn.bind(("127.0.0.1", 9090))
    conn.listen(10)

    while True:
        try:
            clientsocket, address = conn.accept()
            data = clientsocket.recv(1024)
            data.decode('utf-8')
            result = parall_area(data)
            clientsocket.send(result.encode('utf-8'))
            clientsocket.close()
        except KeyboardInterrupt:
            conn.close()
            break
```

```

client.py x
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 9090))
print("Введите длину основания и высоты через пробел:")
values = input()
values = values.encode('utf-8')
conn.send(values)

data = conn.recv(1024)
conn.close()

print(data.decode('utf-8'))

```

Результат:

```

(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task02$ python client.py
Введите длину основания и высоты через пробел:
2 8
16

```

Задание №3

Текст задания:

Реализовать серверную часть приложения. Клиент подключается к серверу. В ответ клиент получает http-сообщение, содержащее html-страницу, которую сервер подгружает из файла index.html.

Код:

```

server.py
import socket

server = socket.socket()
host = '127.0.0.1'
port = 9090
server.bind((host, port))

print('Starting server on', host, port)
print('The Web server URL for this would be http://%s:%d/' % (host, port))

server.listen(5)

print('Entering infinite loop; hit Ctrl+C to exit')
while True:
    client, (client_host, client_port) = server.accept()
    print('Got connction from', client_host, client_port)
    while True:
        data = client.recv(1024)
        if not data:
            break
        response_type = 'HTTP/1.0 200 OK\n'
        headers = 'Content-Type: text/html\n\n'
        with open('index.html', 'r') as f:
            body = f.read()
        response = response_type + headers + body
        client.send(response.encode('utf-8'))
    client.close()

```

```

client.py
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 9090))
conn.send(b"Give Http-message")

data = conn.recv(1024)
udata = data.decode("utf-8")
conn.close()

print(udata)

```

```

index.html
<html>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>

```


Результат:

```
(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task03$ python client.py
HTTP/1.0 200 OK
Content-Type: text/html

<html>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Задание №4

Текст задания:

Реализовать двухпользовательский или многопользовательский чат.

Реализация многопользовательского чата позволяет получить максимальное количество баллов.

Код:

```
server.py
import socket
import threading

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(('', 7070))
conn.listen()

clients = []

print('Start Server')

def new_client():
    while True:
        clientsocket, address = conn.accept()
        if clientsocket not in clients:
            clients.append(clientsocket)
            threading.Thread(target=chat, args=[clientsocket, address]).start()

def chat(clientsocket, address):
    print(address[0], address[1])
    while True:
        try:
            data = clientsocket.recv(1024)

            for client in clients:
                if client == clientsocket:
                    continue
                client.send(data)
        except Exception:
            clients.remove(clientsocket)

    clientsocket.close()

threading.Thread(target=new_client()).start()
```

```

client.py x
import socket
import threading

def chat():
    while True:
        data = conn.recv(1024)
        print(data.decode('utf-8'))

nickname = input()

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(('', 7070))
conn.send(('Please welcome ' + nickname).encode('utf-8'))

threading.Thread(target=chat).start()

while True:
    message = input()
    conn.send(('*' + nickname + '*: ' + message).encode('utf-8'))

```

Результат:

```

(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task04$ python client.py
Jane
*Jake*: hi, Jane
Hi, sis
*Jake*: Periodt
pooh

(base) danil@danil-UX310UQK:~/Education/Web/lab01/Task04$ python client.py
Jake
Please welcome Jane
hi, Jane
*Jane*: Hi, sis
Periodt
*Jane*: pooh

```

Вывод:

В ходе выполнения заданий данной работы были получены практические навыки работы, настройки и реализации web-серверов с использованием сокетов; реализации клиентской и серверной части приложения; реализации двухпользовательского или многопользовательского чата.