

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

Факультет «Инфокоммуникационных технологий»  
Направление подготовки 09.03.03 «Мобильные и сетевые технологии»

**ОТЧЕТ**

---

**Тема задания:** Работа с сокетами

---

**Выполнил:**

Студент: Иванова И.И., К33402  
(ФИО, номер группы)

**Проверил:**

Преподаватель: Говоров А.И.  
(ФИО)

Санкт-Петербург  
2020

**Цель:** овладеть практическими навыками и умениями реализации web-сервер и использования сокетов

### **Практическое задание:**

#### **Часть 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.

Код сервера:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 14900))
conn.listen(10)

while True:
    try:
        clientsocket, address = conn.accept()
        data = clientsocket.recv(106384)
        print(data.decode())
    except KeyboardInterrupt:
        conn.close()
        break
```

Код клиента:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 14900))
conn.send(b"Hello world!")
conn.close()
```

Результат:

```
MacBook-Pro-Inessa:task_1 spisladko$ python3 server.py
Hello world!
█
```

## Часть 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 параллелограмма.

Код сервера:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 14902))
conn.listen(10)

while True:
    clientsocket, address = conn.accept()
    s = int(clientsocket.recv(1024).decode())
    h = int(clientsocket.recv(1024).decode())
    res = str(s*h)
    clientsocket.send(res.encode())
```

Код клиента:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(("127.0.0.1", 14902))
s = input("Введите основание: ")
h = input("Введите высоту: ")
conn.send(s.encode())
conn.send(h.encode())
data = conn.recv(1024)
print(data.decode())
conn.close()
```

Результат:

```
MacBook-Pro-Inessa:task_2 spisladko$ python3 client.py
Введите основание: 5
Введите высоту: 4
20
```

### Часть 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.

Код сервера:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.bind(("127.0.0.1", 14902))
conn.listen(1)

conn, addr = conn.accept()
conn.sendall(b"HTTP/1.0 200 OK\nContent-Type: text/html\n\n" +
open("index.html", "rb").read())

conn.close()
```

Код клиента:

```
import socket

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

conn.connect(("127.0.0.1", 14902))

data = conn.recv(1024)

print(data.decode())

conn.close()
```

Результат:

```
[MBP-Inessa:task_3 spisladko$ python3 client.py
HTTP/1.0 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Задание</title>
</head>
<body>
  <h1>неоднозначное</h1>
</body>
</html>
```

## Часть 4

Код сервера:

```
import socket
import threading
import sys

sock = socket.socket()
sock.bind(("127.0.0.1", 14907))
sock.listen(10)
spisok = []

def recieve():
    while True:
        for connection in spisok:
            try:
                data = connection.recv(1024)
                if data:
                    print(connection, ': ', data.decode())
            except socket.error as e:
                if e.errno == 10053:
                    print("Пользователь:", len(spisok))
                else:
                    raise

def send():
    while True:
        global spisok
        message = input()
        if message:
            for connection in spisok:
                connection.send(message.encode())

def accept():
    while True:
        global spisok
        spisok.append(sock.accept()[0])
        print("Пользователей:", len(spisok))

recieving = threading.Thread(target=recieve)
sending = threading.Thread(target=send)
accepting = threading.Thread(target=accept)
recieving.start()
sending.start()
accepting.start()
```

Код клиента:

```
import socket
import threading
import sys

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(("127.0.0.1", 14907))

def recieve():
    while True:
        data = sock.recv(1024)
        if data:
            print(data.decode())

def send():
    while True:
        message = input()
        if message == "exit":
            sock.close()
            sys.exit()
        else:
            sock.send(message.encode())

recieving = threading.Thread(target=recieve)
sending = threading.Thread(target=send)

recieving.start()
sending.start()
```

Результат:

```
Пользователь: 1
Пользователь: 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 14907), raddr=('127.0.0.1', 64633)> : hi
<socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 14907), raddr=('127.0.0.1', 64634)> : hello
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 14907), raddr=('127.0.0.1', 64633)> : how are you
<socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 14907), raddr=('127.0.0.1', 64634)> : fine thanks
```