

**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", 14915))
conn.listen(10)
clientsocket, addr = conn.accept()
while True:
    data = clientsocket.recv(1024)
    if not data:
        break
    with open('index.html', 'r') as file:
        html = file.read()
        clientsocket.sendall((f'HTTP/1.0 200 OK\nContent-Type:
        text/html\n\n{html}').encode())
conn.close()
```

Часть 4

Код сервера:

```
import socket

from threading import Thread
import time

users = []
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('127.0.0.1', 53331))
sock.listen(10)
sock.setblocking(False)
```

```

def coming_users():
    while True:
        sock.setblocking(True)
        clientsoc, addr = sock.accept()
        sock.setblocking(False)
        if clientsoc not in users:
            users.append((clientsoc, addr))
            print('User connected:', addr)

def message():
    while True:
        text = None
        try:
            for user in users:
                text = user[0].recv(1024).decode('utf-8')
                print('Received text: ' + text)
                if text == "quit":
                    user[0].close()
                    print('User have closed chat: ', user[1])
                    text = f'User {user[1]} have closed chat'
            for send_user in users:
                if send_user[0] == user[0]:
                    continue
                data = f'User {user[1]}: ' + text
                send_user[0].sendall(data.encode('utf8'))
        except socket.error:
            print('waiting...')
            time.sleep(1)
        except KeyboardInterrupt:
            for user in users:
                user[0].close()
            break

```

```
user_thread = Thread(target=coming_users)
message_thread = Thread(target=message)

user_thread.start()
message_thread.start()
```

Код клиента:

```
def send_message():
    while True:
        text = input()
        sock.sendall(bytes(text, "utf-8"))
        if text == "exit":
            sock.close()
            break

def receive_message():
    while True:
        data = sock.recv(1024)
        print(data.decode("utf-8"))

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('localhost', 53331))
send_thread = Thread(target=send_message)
get_thread = Thread(target=receive_message)

send_thread.start()
get_thread.start()
```

Результат можно увидеть в видео: https://www.youtube.com/playlist?list=PLvRG82_4knWZC7ig6oenTNB84-rECxNhS