

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

**Лабораторная работа №1**  
**по дисциплине**  
**«Web-программирование»**

**Выполнил:**  
студент(ка) III курса ИМРиП  
Группы D33101  
Ф.И.О. Ху Эрли

**Проверил:**  
старший реподаватель ФИКТ  
*Говоров Антон Игоревич*

Санкт-Петербург

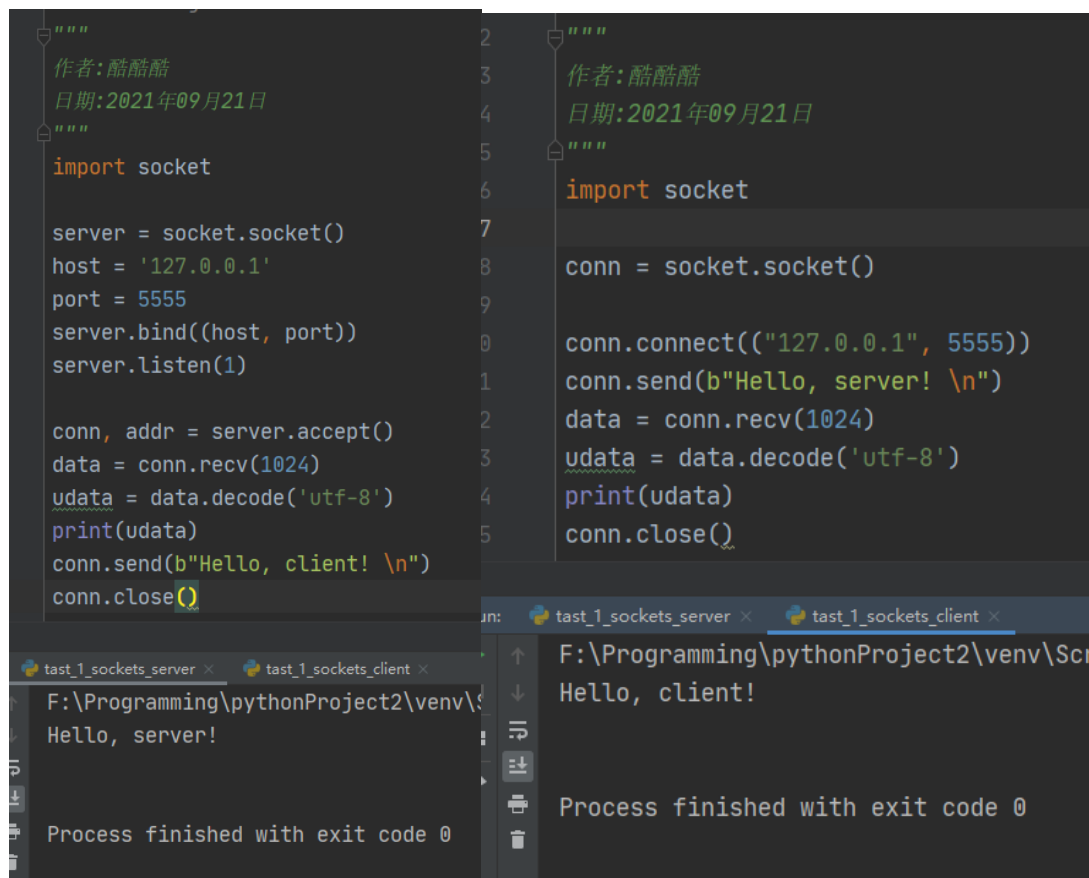
2021

## Работа с сокетами

**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.



```
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket

server = socket.socket()
host = '127.0.0.1'
port = 5555
server.bind((host, port))
server.listen(1)

conn, addr = server.accept()
data = conn.recv(1024)
udata = data.decode('utf-8')
print(udata)
conn.send(b"Hello, client! \n")
conn.close()
```

```
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket

conn = socket.socket()

conn.connect(("127.0.0.1", 5555))
conn.send(b"Hello, server! \n")
data = conn.recv(1024)
udata = data.decode('utf-8')
print(udata)
conn.close()
```

Process finished with exit code 0

Process finished with exit code 0

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. Варианты:

- a. Теорема Пифагораb. Решение квадратного уравнения
- c. Поиск площади трапеции
- d. Поиск площади параллелограмма.

```
# -*- coding:utf-8 -*-
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket
import math

server = socket.socket()
host = '127.0.0.1'
port = 6666
server.bind((host, port))
server.listen(3)

conn, addr = server.accept()
a_bin = conn.recv(1024)
b_bin = conn.recv(1024)
a = a_bin.decode('utf-8')
b = b_bin.decode('utf-8')
c = math.sqrt(int(a)**2 + int(b)**2)
conn.send(str(c).encode())
conn.close()

"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket

conn = socket.socket()

conn.connect(("127.0.0.1", 6666))
a = input("Введите длину первого катета: ")
b = input("Введите длину второго катета: ")
conn.send(a.encode())
conn.send(b.encode())
c_bin = conn.recv(1024)
c = c_bin.decode('utf-8')
print('Гипотенуза треугольника равна: ', c)
conn.close()
```

tast\_2\_sockets\_server x tast\_2\_sockets\_client x

F:\Programming\pythonProject2\venv\Scripts\pyth

Введите длину первого катета: 3

Введите длину второго катета: 4

Гипотенуза треугольника равна: 5.0

Process finished with exit code 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.

```
# -*- coding:utf-8 -*-
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = '127.0.0.1'
port = 7777
server.bind((host, port))
server.listen(1)

conn, addr = server.accept()

page = open('index.html')
content = page.read()
page.close()

response = 'HTTP/1.0 200 OK\n\n' + content
conn.sendall(response.encode())
conn.close()
```

tast\_3\_sockets\_client x tast\_3\_sockets\_server x

F:\Programming\pythonProject2\venv\Scripts\python.exe H:/代码

Process finished with exit code 0

```
# -*- coding:utf-8 -*-
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket

conn = socket.socket()

conn.connect(("127.0.0.1", 7777))
result = conn.recv(10240)
print(result.decode())
conn.close()
```

tast\_3\_sockets\_client x tast\_3\_sockets\_server x

F:\Programming\pythonProject2\venv\Scripts\python.exe H:/代码/tast\_3\_sockets\_client.py

HTTP/1.0 200 OK

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>A tiny tiny page</title>

</head>

<body>

    <p>

        "Go for someone who makes you smile because it takes only a smile to make a dark day seem bright."

    </p>

</body>

</html>

Process finished with exit code 0

#### 4. Реализовать двухпользовательский или многопользовательский чат.

Реализация многопользовательского чата позволяет получить

максимальное количество баллов.

Server:

```
# -*- coding:utf-8 -*-
"""
作者: 酷酷酷
日期: 2021年09月21日
"""
import socket, threading

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Создание сокетов. IPV4, TCP
host = '127.0.0.1'
port = 8888 #Адрес
server.bind((host, port)) #Привязка адреса и порта
server.listen() #Настройка слушателя

clients = []
users = []

def broadcast(msg, client):
    for each in clients:
        if each != client: #Отправьте сообщение каждому второму клиенту
            each.send(msg)

def handle(client):
    while True:
        msg = client.recv(2048) #Получение информации
        broadcast(msg, client)

def receive():
    while True:
        client, addr = server.accept() # Принимает запрос на соединение от клиента и возвращает новый сокет и адрес
        client.send('username'.encode()) # Отправить байтовую строку 'username'
        user = client.recv(2048).decode() # Получение байтовых строк
        clients.append(client) # Добавление клиентов в список клиентов
        users.append(user) # Добавить пользователя в список пользователей
        client.send('Connection established'.encode()) # Отправка байтовых строк
        thread = threading.Thread(target=handle, args=(client,)) #Внедрение многопоточности
        thread.start() #Включение многопоточности

receive()
```

task\_4\_sockets\_server x task\_4\_sockets\_client\_2 x task\_4\_sockets\_client x

F:\Programming\pythonProject2\venv\Scripts\python.exe H:/代码/task\_4\_sockets\_server.py

Client:

```
1  # -*- coding:utf-8 -*-
2  """
3  作者: 酷酷酷
4  日期: 2021年09月21日
5  """
6  import socket
7  import threading  #Импорт многопоточности
8
9
10 conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Создание сокетов. IPV4. TCP
11 server = '127.0.0.1', 8888 #Адрес
12 conn.connect(server) #Подключение к серверу
13
14 username = input('Выберите псевдоним: ')
15
16
17 def recv_msg():
18     while True:
19         msg = conn.recv(2048).decode() #Получение информации
20         if msg == 'username':
21             conn.send(username.encode()) #Получение информации
22         else:
23             print(msg)
24
25
26 def print_msg():
27     while True:
28         msg = '{} says: {}'.format(username, input('')) #Модуль, который продолжает закидывать это входное сообщение и отправляет его
29         conn.send(msg.encode())
30
31
32 recv_thr = threading.Thread(target=recv_msg) #Instantiation
33 print_thr = threading.Thread(target=print_msg)
34 recv_thr.start() #Включение многопоточности
35 print_thr.start() #Включение многопоточности
```

Chatroom:

```
n:  tast_4_sockets_server x  tast_4_sockets_client_2 x  tast_4_sockets_client x
↑  F:\Programming\pythonProject2\venv\Scripts\python.exe
↓  Выберите псевдоним: wux
↻  Connection established
⇅  nihao
🖨  huerli says: nihao
🗑  wobuhao
    huerli says: najiuhao
```

```
tast_4_sockets_server × tast_4_sockets_client_2 × tast_4_sockets_client ×
F:\Programming\pythonProject2\venv\Scripts\python.exe H:/代码/tas
Выберите псевдоним: huerli
Connection established
wux says: nihao
nihao
wux says: wobuhao
najiuhao
```