

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Кафедра Интеллектуальных систем в гуманитарной сфере,  
факультет Инфокоммуникационных технологий

**Лабораторная работа №1**

**Работа с сокетами**

Выполнил

студент гр. № K3242

Кузьмичев Кирилл Максимович

Проверил преподаватель

Говоров А.И.

## Задание №1

**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.

### 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:

```
Client1.py x
1  import socket
2
3  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  Sock.connect(("127.0.0.3", 11042))
5  Sock.send(b"Hello, server! \n")
6
7  data = Sock.recv(1024)
8  Sock.close()
9
10 print(data.decode("utf-8"))
```

```
Server1.py x
1  import socket
2
3  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  Sock.bind(("127.0.0.3", 11042))
5  Sock.listen(1)
6  conn, addr = Sock.accept()
7
8  while True:
9      data = conn.recv(1024)
10     if not data:
11         break
12     print(data.decode("utf-8"))
13     conn.send(b'Hello back!')
14
15     conn.close()
```

## Результат:

```
Server1 x
C:\Users\kirmr\PycharmProjects\
Hello, server!
```

```
Client1 x
C:\Users\kirmr\PycharmProjects\
Hello back!
```

## 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: 3; с. Поиск площади трапеции.)

### Код:

```
Client2.py x
1  import socket
2
3  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  sock.connect(('127.0.0.1', 8888))
5
6  while True:
7      data = sock.recv(1024)
8      if not data:
9          sock.close()
10         break
11     print(data.decode("utf-8"))
12     sock.sendall(bytes(input(), "utf-8"))
13
14 sock.close()
```

```

Server2.py x
1 import socket
2 from math import sqrt
3
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 sock.bind(("127.0.0.1", 8888))
6 sock.listen(1)
7 conn, addr = sock.accept()
8
9 print('connected:', addr, conn)
10 d = ''
11 option = ''
12 while True:
13     conn.sendall(bytes(d + f'Каким способом хотите найти площадь трапеции: \n (a) - Через основания и высоту.
14     data = conn.recv(1024)
15     option = data.decode()
16
17     if option == 'a':
18         conn.sendall(bytes(d + f'Введите значения двух оснований(a, b) и высоты(h): ', "utf-8"))
19         data = conn.recv(1024)
20         print(data.decode())
21         a, b, h = data.decode("utf-8").split(' ')
22         S = (int(h) * (int(a)+int(b)))/2
23         d = f'Площадь равна = {S} \n'
24
25     if option == 'b':
26         conn.sendall(bytes(d + f'Введите значения средней линии(m) и высоты(h): ', "utf-8"))
27         data = conn.recv(1024)
28         print(data.decode())
29         m, h = data.decode("utf-8").split(' ')
30         S = int(m) * int(h)
31         d = f'Площадь равна = {S} \n'
32
33     if not data:
34         break
35     sock.close()

```

## Результат:

```

Server2 x Client2 x
C:\Users\kirmr\PycharmProjects\pythonProject\venv\Scripts\python.exe
Каким способом хотите найти площадь трапеции:
(a) - Через основания и высоту, (b) - Через среднюю линию и высоту
a
Введите значения двух оснований(a, b) и высоты(h):
10 5 5
Площадь равна = 37.5

```

```

Server2 x Client2 x
C:\Users\kirmr\PycharmProjects\
connected: ('127.0.0.1', 64448)
10 5 5

```

### 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:

```
Client3.py x
1  import socket
2
3  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  Sock.connect(("127.0.0.1", 11042))
5  Sock.send('Hello, server!'.encode("utf-8"))
6
7  data = Sock.recv(1024)
8  Sock.close()
9
10 print(data.decode("utf-8"))
```

```
Server3.py x
1  import socket
2
3  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  Sock.bind(("127.0.0.1", 11042))
5  Sock.listen(1)
6  conn, addr = Sock.accept()
7
8  while True:
9      with open('index.html', 'r') as file:
10         browse = file.read()
11         conn.sendall(bytes(f'{browse}', 'utf-8'))
12         data = conn.recv(1024)
13         if not data:
14             break
15
16  conn.close()
```

```
index.html x
1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <title>Hola, soi Dora!</title>
6      </head>
7      <body>
8
9      </body>
10     </html>
```

**Результат:**

```
Server3 x Client3 x
C:\Users\kirmr\PycharmProjects\pythonProject
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hola, soi Dora!</title>
</head>
<body>

</body>
</html>
```

## Задание №4

### Текст задания:

Реализовать двух пользовательский или многопользовательский чат. Реализация многопользовательского чата позволяет получить максимальное количество баллов.

### Код:

```
Client4_1.py x
1  import socket
2  import threading
3
4  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5  Sock.connect(("127.0.0.2", 11042))
6
7  def send():
8      while True:
9          text = input()
10         Sock.send(b"Kirill: " + text.encode("utf-8"))
11
12  def get():
13      while True:
14          feedback = Sock.recv(1024)
15          print(feedback.decode("utf-8"))
16
17  threading.Thread(target = send).start()
18  threading.Thread(target = get).start()
```

Client4\_2.py ×

```
1  import socket
2  import threading
3
4  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5  Sock.connect(("127.0.0.2", 11042))
6
7  def send():
8      while True:
9          text = input()
10         Sock.send(b"Polina: " + text.encode("utf-8"))
11
12  def get():
13      while True:
14          feedback = Sock.recv(1024)
15          print(feedback.decode("utf-8"))
16
17  threading.Thread(target = send).start()
18  threading.Thread(target = get).start()
```

Client4\_3.py ×

```
1  import socket
2  import threading
3
4  Sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5  Sock.connect(("127.0.0.2", 11042))
6
7  def send():
8      while True:
9          text = input()
10         Sock.send(b"Daniil: " + text.encode("utf-8"))
11
12  def get():
13      while True:
14          feedback = Sock.recv(1024)
15          print(feedback.decode("utf-8"))
16
17  threading.Thread(target = send).start()
18  threading.Thread(target = get).start()
```



```

Server4.py x
2  import threading
3
4  S = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5  S.bind(("127.0.0.2", 11042))
6  S.listen(3)
7  Account = []
8
9  def add_clients():
10     while True:
11         Sock, addr = S.accept()
12         Account.append(Sock)
13         threading.Thread(target=_chat, args=_[Sock, addr]).start()
14
15  def chat(Sock, addr):
16     print(addr)
17     while True:
18         try:
19             feedback = Sock.recv(1024)
20             if not feedback:
21                 break
22             for client in Account:
23                 if client == Sock:
24                     continue
25                 client.sendall(feedback)
26             except Exception:
27                 Account.remove(Sock)
28                 break
29         Sock.close()
30
31  threading.Thread(target=add_clients()).start()

```

Результат:

```

Server4 x Client4_1 x Client4_2 x Client4_3 x
C:\Users\kirmr\PycharmProjects\pythonProject\venv
('127.0.0.1', 64476)
('127.0.0.1', 64477)
('127.0.0.1', 64484)

```

```
Server4 x Client4_1 x Client4_2 x Client4_3 x
C:\Users\kirmr\PycharmProjects\pythonProject\venv
Привет
Polina: Hola
Danil: Hello

Server4 x Client4_1 x Client4_2 x Client4_3 x
C:\Users\kirmr\PycharmProjects\pythonProject\venv
Kirill: Привет
Hola
Danil: Hello

Server4 x Client4_1 x Client4_2 x Client4_3 x
C:\Users\kirmr\PycharmProjects\pythonProject\venv
Kirill: Привет
Polina: Hola
Hello
```

## 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.