

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ**

**ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»  
Кафедра «Интеллектуальных технологий в гуманитарной сфере»  
Направление подготовки «09.03.03 Прикладная информатика»

## **ОТЧЕТ**

### **По лабораторной работе №1**

---

**Тема задания** Работа с сокетами (Вариант 10(2))

---

**Выполнил:**

**Студент** **Никитин М. М.** **К33401**  
(Фамилия И.О.) номер группы

**Санкт-Петербург  
2021**

## Цель работы

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.
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. Вариант: Поиск площади трапеции.
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.
4. Реализовать двухпользовательский или многопользовательский чат. Реализация многопользовательского чата позволяет получить максимальное количество баллов.

# Выполнение

## Задание №1

Реализация серверной и клиентской части приложения.

Реализация в программном коде

Серверная часть:

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind(('127.0.0.1', 14900))
sock.listen(1)
cl_sock, addr = sock.accept()
print('connected:', addr)

while True:
    data = cl_sock.recv(16384)
    if not data:
        break
    print(data.decode("utf-8"))
    cl_sock.sendall(b'Hello, client!')

cl_sock.close()
```

Клиентская часть:

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1', 14900))
sock.sendall(b'Hello, server!')

data = sock.recv(16384)
print(data.decode("utf-8"))

sock.close()
```

Результат работы со стороны сервера:

```
C:\Users\User\Desktop\ITMO_ICT_Web
connected: ('127.0.0.1', 51451)
Hello, server!

Process finished with exit code 0
```

Результат работы со стороны клиента:

```
C:\Users\User\Desktop\ITMO_ICT_Web
Hello, client!

Process finished with exit code 0
```

## Задание №2

Реализация серверной и клиентской части приложения. Математические операции.

Реализация в программном коде

Серверная часть:

```
1  import socket
2  import math
3
4  def sqr_solution(a, b, c):
5      D = b * b - (4 * a * c)
6      if D < 0:
7          result = f"no roots\n"
8      elif D == 0:
9          x = (-1 * b) / (2 * a)
10         result = f"1 root    x={x} \n"
11     else:
12         D = math.sqrt(D)
13         x = (-1 * b - D) / (2 * a)
14         x2 = (-1 * b + D) / (2 * a)
15         result = f"2 roots    x1={x}, x2={x2} \n"
16     return result
17
18     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
19     sock.bind(('127.0.0.1', 14900))
20     sock.listen(1)
21     cl_sock, addr = sock.accept()
22     cl_sock.send(bytes(f'enter coefficients "a", "b", "c" of equation "ax^2 + bx + c = y" \n '
23                       f'or enter "exit" to finish the program', "utf-8"))
24     print('connected:', addr)
25
26     while True:
27         data = cl_sock.recv(16384)
28         if not data:
29             break
30         a, b, c = data.decode("utf-8").split(' ')
31         result = sqr_solution(int(a), int(b), int(c))
32         cl_sock.send(str(result).encode("UTF-8"))
33
34     cl_sock.close()
```

Клиентская часть:

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(('127.0.0.1', 14900))

data = sock.recv(16384)
print(data.decode("UTF-8"))
abc = input()
sock.send(abc.encode("UTF-8"))
data = sock.recv(16384)
print(data.decode("UTF-8"))
sock.close()
```

Результат работы со стороны сервера:

```
C:\Users\User\Desktop\ITMO_ICT_WebDev
connected: ('127.0.0.1', 51709)

Process finished with exit code 0
```

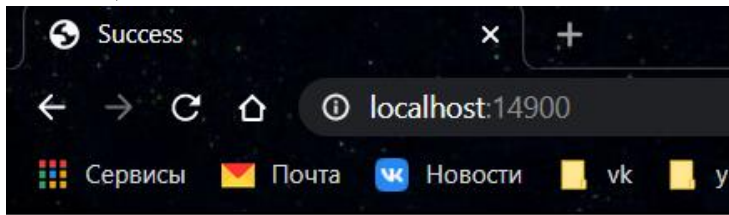
Результат работы со стороны клиента:

```
C:\Users\User\Desktop\ITMO_ICT_WebDevelopment_2020-2021\students\K3
enter coefficients "a", "b", "c" of equation "ax^2 + bx + c = y"
or enter "exit" to finish the program
1 4 4
1 root    x=-2.0

Process finished with exit code 0
```

## Задание №3

Реализация серверной и клиентской части приложения. Получение http-сообщения.



## You receive free message

Реализация в программном коде

Серверная часть:

```
1 import socket
2
3 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 sock.bind(('127.0.0.1', 14900))
5 sock.listen(1)
6 cl_sock, addr = sock.accept()
7 print('connected:', addr)
8 with open('index.html', 'r') as file:
9     html = file.read()
10    cl_sock.sendall(bytes(f'HTTP/1.0 200 OK\nContent-Type: text/html\n\n{html}', 'utf-8'))
11
12 cl_sock.close()
```

Клиентская часть:

```
1 import socket
2
3
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 sock.connect(('127.0.0.1', 14900))
6
7 while True:
8     data = sock.recv(16384)
9     if not data:
10        break
11    print(data.decode("utf-8"))
12
13 sock.close()
```

Результат работы со стороны сервера:

```
C:\Users\User\Desktop\ITMO_ICT_WebDevelopm
connected: ('127.0.0.1', 52369)

Process finished with exit code 0
```



Результат работы со стороны клиента:

```
C:\Users\User\Desktop\ITMO ICT WebDevelopment_2
HTTP/1.0 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Success</title>
</head>
<body>
  <h1>You receive free message</h1>
</body>
</html>

Process finished with exit code 0
```



## Задание №4

Реализация многопользовательского чата.

Реализация в программном коде

Серверная часть:

```
1 import socket
2 from threading import Thread
3
4
5 sock = socket.socket()
6 sock.bind(('127.0.0.1', 53331))
7 sock.listen(10)
8 users = []
9
10
11 def adder():
12     while True:
13         global users
14         users.append(sock.accept()[0])
15         print("Пользователь подключился, стало:", len(users))
16
17
18 def users_counter():
19     while True:
20         for i in range(len(users)):
21             try:
22                 data = users[i].recv(4096)
23                 if data:
24                     message = data.decode()
25                     print(message)
26                     if message == "exit":
27                         users.pop(i)
28                     print("Пользователь отключился, осталось:", len(users))
29             except socket.error as error:
30                 if error.errno == 10054:
31                     users.pop(i)
32                     print("Пользователь отключился, осталось:", len(users))
33             else:
34                 raise
35
36
37 def messenger():
38     while True:
39         global users
40         message = input()
41         if message:
42             for i in range(len(users)):
43                 users[i].send(message.encode())
44
45
46 user_thread = Thread(target=users_counter)
47 message_thread = Thread(target=messenger)
48 adder_thread = Thread(target=adder)
49
50
51 user_thread.start()
52 message_thread.start()
53 adder_thread.start()
```

## Клиентская часть:

```
1 import socket, sys
2 from threading import Thread
3
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 sock.connect(('127.0.0.1', 53331))
6 print("Connected to chat")
7
8
9 def messenger():
10     while True:
11         message = input()
12         if message == "exit":
13             sock.send(message.encode())
14             sock.close()
15             sys.exit()
16         else:
17             sock.send(message.encode())
18
19
20 def receiver():
21     while True:
22         data = sock.recv(4096)
23         if data:
24             print(data.decode())
25
26
27 message_thread = Thread(target=messenger)
28 get_thread = Thread(target=receiver)
29
30 message_thread.start()
31 get_thread.start()
```

## Результат работы со стороны сервера:

```
C:\Users\User\Desktop\ITMO_ICT_WebDevelopment
Пользователь подключился, стало: 1
Пользователь подключился, стало: 2
Пользователь подключился, стало: 3
privet
howdy
hi
priv
zdarova
Good morning
Пользователь отключился, осталось: 2
|
```

## Результат работы со стороны клиентов:

```
C:\Users\User\Desktop\ITMO_ICT_WebDeve
Connected to chat
privet
howdy
hi
Good morning

Process finished with exit code -1
|
```