

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Отчет

По лабораторной работе №1

«Веб-программирование»

Выполнила:
Герулайтите Габриэля

Группа:
К33401

Санкт-Петербург
2021 г.

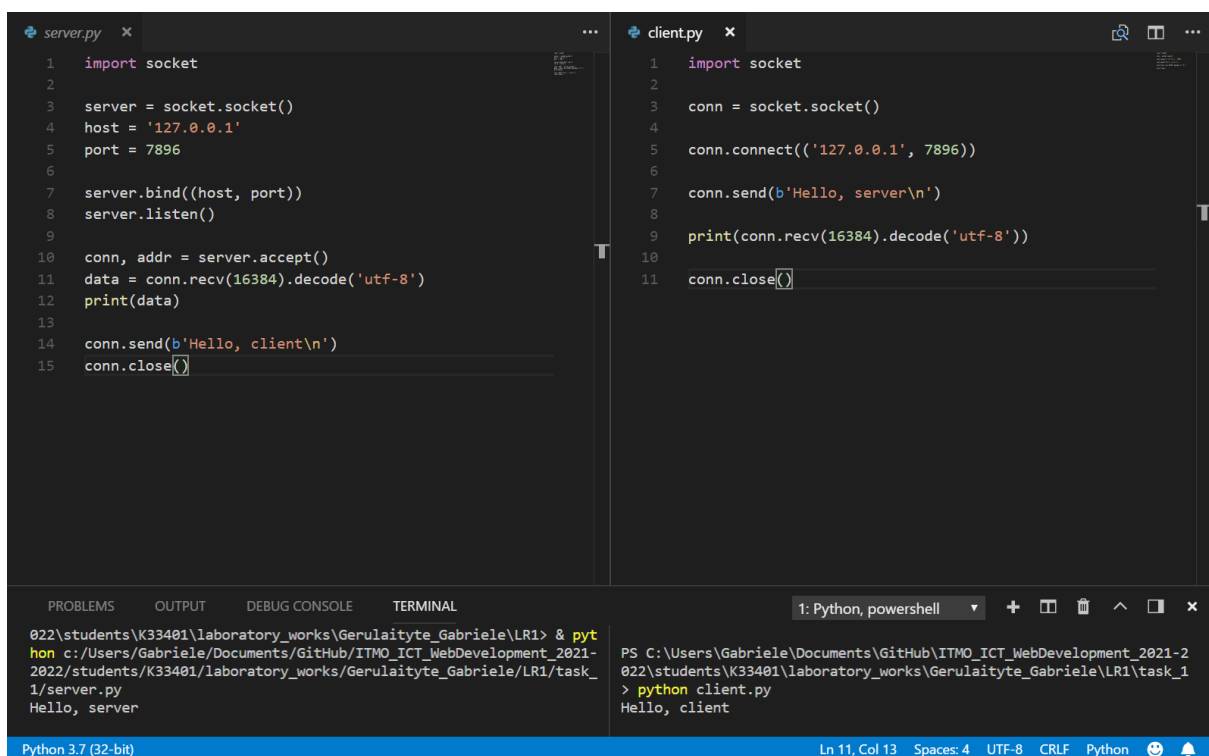
Цель работы:

овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.

Выполнение работы:

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.



```
server.py
1 import socket
2
3 server = socket.socket()
4 host = '127.0.0.1'
5 port = 7896
6
7 server.bind((host, port))
8 server.listen()
9
10 conn, addr = server.accept()
11 data = conn.recv(16384).decode('utf-8')
12 print(data)
13
14 conn.send(b'Hello, client\n')
15 conn.close()

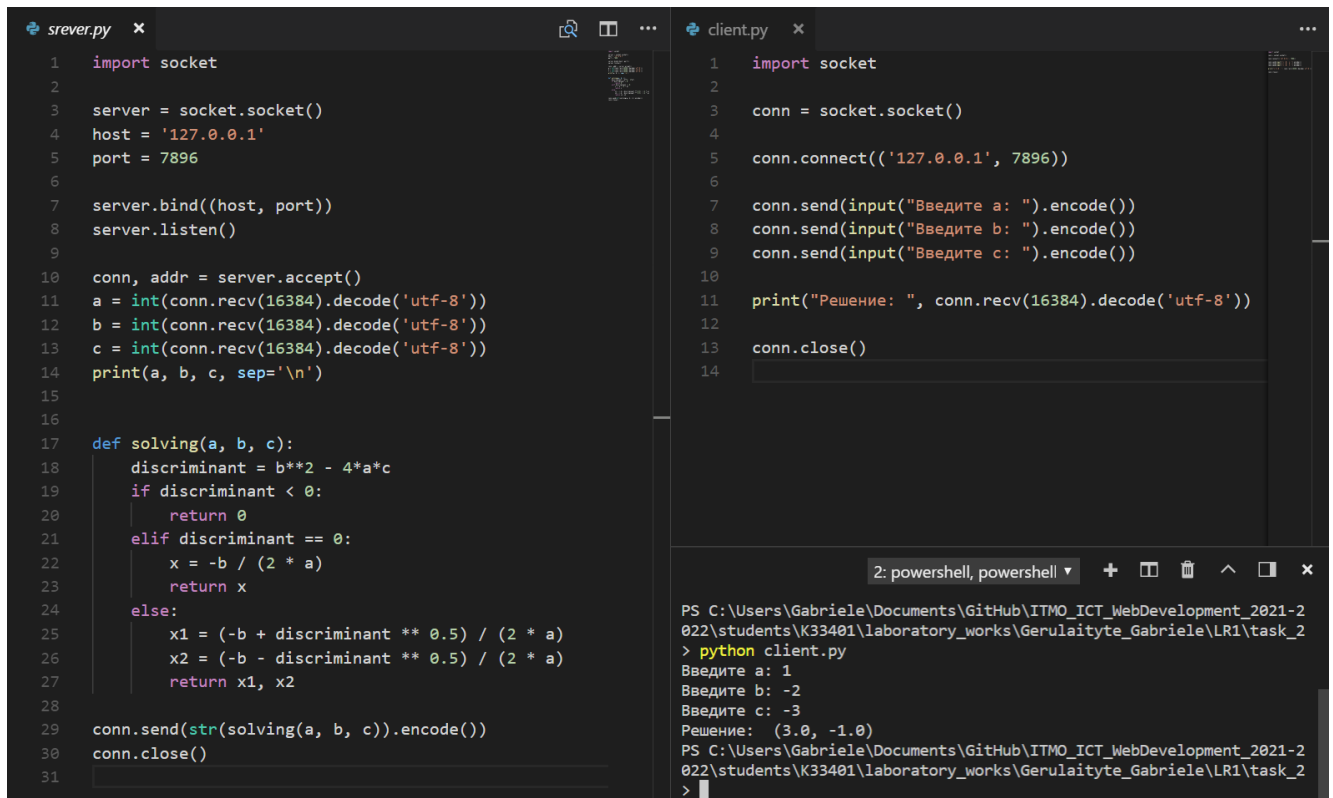
client.py
1 import socket
2
3 conn = socket.socket()
4
5 conn.connect(('127.0.0.1', 7896))
6
7 conn.send(b'Hello, server\n')
8
9 print(conn.recv(16384).decode('utf-8'))
10
11 conn.close()

Terminal
022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1> & pyth
hon c:/Users/Gabriele/Documents/GitHub/ITMO ICT_WebDevelopment_2021-2
2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_
1/server.py
Hello, server

PS C:\Users\Gabriele\Documents\GitHub\ITMO ICT_WebDevelopment_2021-2
022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_1
> python client.py
Hello, client
```

Практическое задание 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: Решение квадратного уравнения.



```
srever.py x
1 import socket
2
3 server = socket.socket()
4 host = '127.0.0.1'
5 port = 7896
6
7 server.bind((host, port))
8 server.listen()
9
10 conn, addr = server.accept()
11 a = int(conn.recv(16384).decode('utf-8'))
12 b = int(conn.recv(16384).decode('utf-8'))
13 c = int(conn.recv(16384).decode('utf-8'))
14 print(a, b, c, sep='\n')
15
16
17 def solving(a, b, c):
18     discriminant = b**2 - 4*a*c
19     if discriminant < 0:
20         return 0
21     elif discriminant == 0:
22         x = -b / (2 * a)
23         return x
24     else:
25         x1 = (-b + discriminant ** 0.5) / (2 * a)
26         x2 = (-b - discriminant ** 0.5) / (2 * a)
27         return x1, x2
28
29 conn.send(str(solving(a, b, c)).encode())
30 conn.close()
31

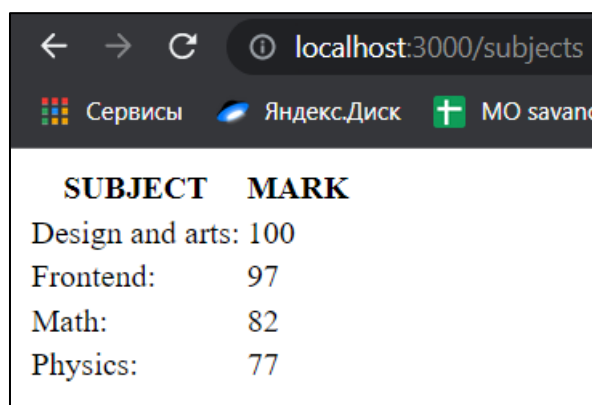
client.py x
1 import socket
2
3 conn = socket.socket()
4
5 conn.connect(('127.0.0.1', 7896))
6
7 conn.send(input("Введите a: ").encode())
8 conn.send(input("Введите b: ").encode())
9 conn.send(input("Введите c: ").encode())
10
11 print("Решение: ", conn.recv(16384).decode('utf-8'))
12
13 conn.close()
14

2: powershell, powershell
PS C:\Users\Gabriele\Documents\GitHub\ITMO ICT WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_2> python client.py
Введите a: 1
Введите b: -2
Введите c: -3
Решение: (3.0, -1.0)
PS C:\Users\Gabriele\Documents\GitHub\ITMO ICT WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_2>
```

Исправно работает в случаях, когда дискриминант больше, меньше или равен 0.

Практическое задание 3:

Необходимо написать простой web-сервер для обработки GET и POST http запросов средствами Python и библиотеки socket.



| SUBJECT | MARK |
|------------------|------|
| Design and arts: | 100 |
| Frontend: | 97 |
| Math: | 82 |
| Physics: | 77 |

Практическое задание 4:

Реализовать двухпользовательский или многопользовательский чат.

Реализация многопользовательского чата позволяет получить максимальное количество баллов.

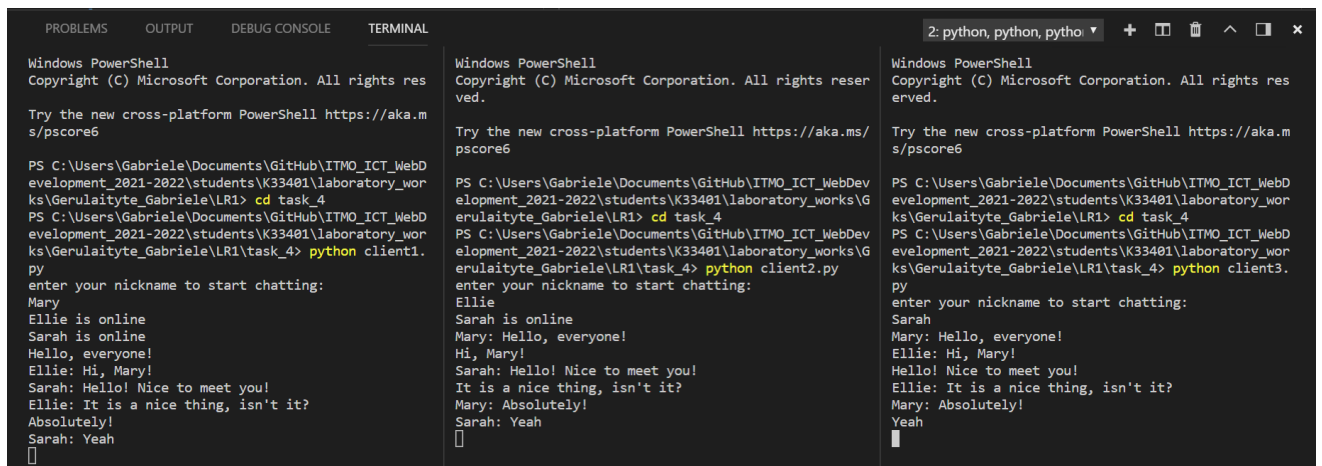
```
server.py x
1 import socket
2
3
4 class Server:
5     def __init__(self, port: int):
6         self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
7         self.sock.bind(('127.0.0.1', port))
8
9     def run(self):
10        clients = []
11        while True:
12            data, address = self.sock.recvfrom(1024)
13            print(address[0], address[1])
14            if address not in clients:
15                clients.append(address)
16            for client in clients:
17                if client == address:
18                    continue
19                self.sock.sendto(data, client)
20
21
22 if __name__ == '__main__':
23     server = Server(14900)
24     server.run()
```

```
client1.py x client2.py
1 import socket
2 import threading
3
4
5 class Client:
6     def __init__(self, port: int):
7         self.server = '127.0.0.1'
8         self.conn = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9         self.conn.bind((self.server, port))
10
11     def read(self):
12         while True:
13             data = self.conn.recv(1024)
14             print(data.decode('utf-8'))
15
16     def chat(self):
17         print("enter your nickname to start chatting: ")
18         nickname = input()
19         self.conn.sendto((nickname + ' is online').encode('utf-8'),
20                           (self.server, self.conn.getsockname()[1]))
21         while True:
22             self.conn.sendto((nickname + ': ' + input()).encode('utf-8'),
23                               (self.server, self.conn.getsockname()[1]))
24
25
26 if __name__ == '__main__':
27     client = Client(14900)
28     thread1, thread2 = threading.Thread(target=client.read), threading.Thread(target=client.chat)
29     thread1.start(), thread2.start()
```

```
client1.py x client2.py client3.py
1 import socket
2 import threading
3
4
5 class Client:
6     def __init__(self, port: int):
7         self.server = '127.0.0.1', port
8         self.conn = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9         self.conn.bind((self.server, 0))
10
11     def read(self):
12         while True:
13             data = self.conn.recv(1024)
14             print(data.decode('utf-8'))
15
16     def chat(self):
17         print("enter your nickname to start chatting: ")
18         nickname = input()
19         self.conn.sendto((nickname + ' is online').encode('utf-8'),
20                           (self.server, self.conn.getsockname()[1]))
21         while True:
22             self.conn.sendto((nickname + ': ' + input()).encode('utf-8'),
23                               (self.server, self.conn.getsockname()[1]))
24
25
26 if __name__ == '__main__':
27     client = Client(14900)
28     thread1, thread2 = threading.Thread(target=client.read), threading.Thread(target=client.chat)
29     thread1.start(), thread2.start()
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

Чат:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1> cd task_4
PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_4> python client1.py
enter your nickname to start chatting:
Mary
Ellie is online
Sarah is online
Hello, everyone!
Ellie: Hi, Mary!
Sarah: Hello! Nice to meet you!
Ellie: It is a nice thing, isn't it?
Absolutely!
Sarah: Yeah
█

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1> cd task_4
PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_4> python client2.py
enter your nickname to start chatting:
Ellie
Sarah is online
Mary: Hello, everyone!
Hi, Mary!
Sarah: Hello! Nice to meet you!
It is a nice thing, isn't it?
Mary: Absolutely!
Sarah: Yeah
█

2: python, python, python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1> cd task_4
PS C:\Users\Gabriele\Documents\GitHub\ITMO_ICT_WebDevelopment_2021-2022\students\K33401\laboratory_works\Gerulaityte_Gabriele\LR1\task_4> python client3.py
enter your nickname to start chatting:
Sarah
Mary: Hello, everyone!
Ellie: Hi, Mary!
Hello! Nice to meet you!
Ellie: It is a nice thing, isn't it?
Mary: Absolutely!
Yeah
█
```

Выводы:

На примере библиотеки socket языка Python было изучено устройство web-серверов и на практике реализованы их примеры.