**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**РАБОТА С СОКЕТАМИ**

РАБОТУ ВЫПОЛНИЛА:

Штрейх Анна Андреевна

РАБОТУ ПРОВЕРИЛ:

Говоров Антон Игоревич

Санкт-Петербург

2021 г.

**Цель работы:**

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

**Выполнение:**

1. Реализуем клиентскую и серверную часть приложения. Клиент и сервер отсылают друг другу приветствие.

Создается сокет, который связывается с определенным хостом и портом. Затем для сокета запускается режим прослушивания, принимается подключение с новым сокетом и адресом клиента. Получаем данные, дешифруем их и выводим. Чтобы отправить – шифруем. В конце со закрывается.

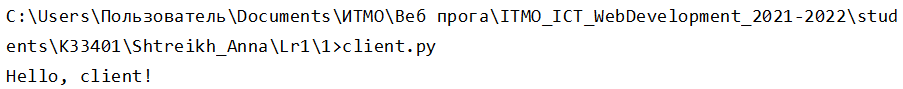
Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 1 – server.py

Рисунок 2 – client.py



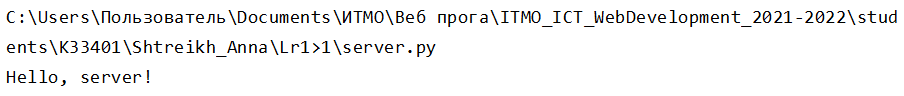
Рисунок 3 – Работа программы на клиенте

Рисунок 4 – Работа программы на сервере

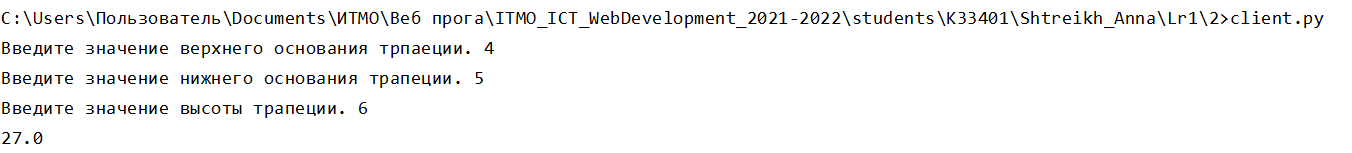
1. Теперь реализуем клиентскую и серверную часть так, чтобы сервер решал математическую задачу, принятую от клиента.

Рисунок 5 – server.py

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – client.py

Рисунок 7 – Работа программы на клиенте

1. Реализуем работу web-сервера для обработки GET и POST http запросов.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – Импортируем необходимое

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – Базовый класс для работы сервера

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 – функция serve\_client

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 11 – функция parse\_request

Рисунок 12 – функция parse\_request\_line

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 13 – функция parse\_headers

Рисунок 14 – функция handle\_request

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 – функция handle\_post\_mark

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 – функция handle\_get\_marks

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17 – функция handle\_get\_sub

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18 – функция send\_response

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19 – функция send\_error

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 20 – класс Request

Рисунок 21 – функция body в классе Request

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22 – классы Response и HTTPError

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23 – Вызов работы сервера

Для проверки работы сервера воспользуемся Postman.

Сделаем несколько пост-запросов с разными предметами и оценками

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 24 – ввод предмета и оценки

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 25 – вывод оценок по всем добавленным предметам

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 26 – вывод оценок по конкретному предмету

1. Изображение выглядит как текст

   Автоматически созданное описаниеРеализуем многопользовательский чат с использованием потоков.

Рисунок 27 – файл server.py начало

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 28 – файл server.py функции

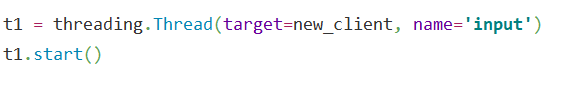


Рисунок 29 – запуск потока

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 30 – файл client.py [1]

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 31 – файл client.py[2]

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 32 – запуск клиент 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 33 – запуск клиент 2

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 34 – запуск клиент 3

**Вывод:**

Получены базовые навыки, необходимые для реализации web-серверов и использования сокетов.