**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по дисциплине «Теория языков программирования»

Выполнил:

студент группы БСТ1801

Коловертных Е.С.

Москва 2019

1. Цель работыпостроение клиент-серверного приложения с использованием ЯП Python.
2. Задания
   1. Первое задание:

На запрос клиента сервер возвращает текущую дату и время.

* 1. Второе задание:

Сервер принимает от клиента значения переменных, необходимые для решения уравнения своего варианта 1 задания лабораторной работы №1, в порядке вхождения в уравнение и возвращает результат вычислений или сообщение об ошибке деления на 0.

* 1. Третье задание:

Сервер принимает от клиента матрицу размера 3х3 и возвращает вычисленный определитель.

* 1. Четвертое задание:

Сервер запоминает входящее текстовое сообщение клиента и возвращает клиенту предыдущее сохраненное сообщение. На первый после запуска сервера запрос от любого клиента возвращается строка «Поступившего ранее сообщения не было»

* 1. Пятое задание:

Сервер принимает от клиента матрицу размера 2х2 и возвращает матрицу, элементы которой увеличены в det A раз.

1. Ход работы
   1. Код приложения клиента первой программы

import socket

sock = socket.socket()

sock.connect(('127.0.0.1', 9090))

message = input('Введите что-нибудь: \n')

sock.send(message.encode())

print(sock.recv(1024).decode())

sock.close()

* 1. Код приложения сервера первой программы

from time import localtime, strftime

import socket

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

if len(conn.recv(1024)) > 0:

conn.send(strftime("%d.%m.%Y %H:%M:%S", localtime()).encode())

sock.close()

* 1. Скриншот выполнения программ первого задания

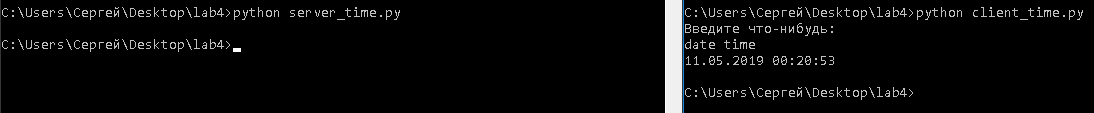
Скриншот выполнения программы представлен на рисунке 1  


Рисунок 1 - Результат выполнения программ первого задания

* 1. Код клиента второго задания

import socket

sock = socket.socket()

sock.connect(('127.0.0.1', 9090))

user\_message = input('Введите значения через пробел \n')

sock.send(user\_message.encode())

print(sock.recv(1024).decode())

* 1. Код сервера второй программы

import socket

def solver(data):

a, b, c, k = data[0], data[1], data[2], data[3]

if b == 0 or a == 0 or (a + b + c \* (k - a / b\*\*3)) == 0:

return "Деление на ноль невозможно :("

else:

return abs((a\*\*2 / b\*\*2 + c\*\*2 \* a\*\*2)/(a + b + c \* (k - a / b\*\*3)) + c + c \* (k/b - k/a))

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

data = conn.recv(1024).decode().strip().split(' ')

data = [int(x) for x in data]

answer = str(solver(data))

conn.send(answer.encode())

sock.close()

* 1. Скриншот выполнения программ второго задания

Скриншот выполнения программы представлен на рисунке 2

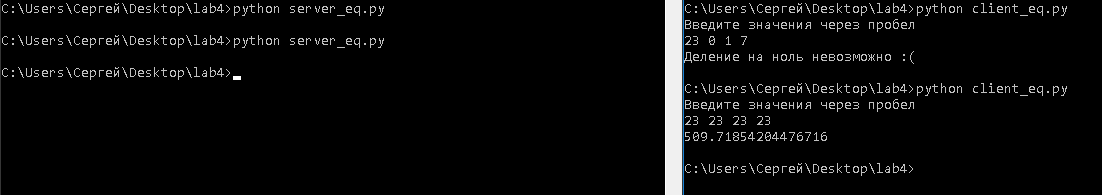


Рисунок 2- Результат выполнения второй программы

* 1. Код клиента третьего задания

import socket

user\_message = ''

sock = socket.socket()

sock.connect(('127.0.0.1', 9090))

for i in range(0,3):

user\_message += input(f'Введите {i+1}-ю строку: ') + ' '

sock.send(user\_message.encode())

data = sock.recv(1024)

print(data.decode())

sock.close()

* 1. Код сервера третьего задания

import socket

from datetime import datetime

def det(data):

return (data[0] \* (data[4] \* data[8] - data[5] \* data[7]) -

data[1] \* (data[3] \* data[8] - data[5] \* data[6]) +

data[2] \* (data[3] \* data[7] - data[6] \* data[4]))

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

data = conn.recv(1024).decode().strip().split(' ')

data = [int(x) for x in data]

answer = str(det(data))

conn.send(answer.encode())

sock.close()

* 1. Скриншот выполнения программ третьего задания

Скриншот выполнения программ третьего задания представлен на рисунке 3

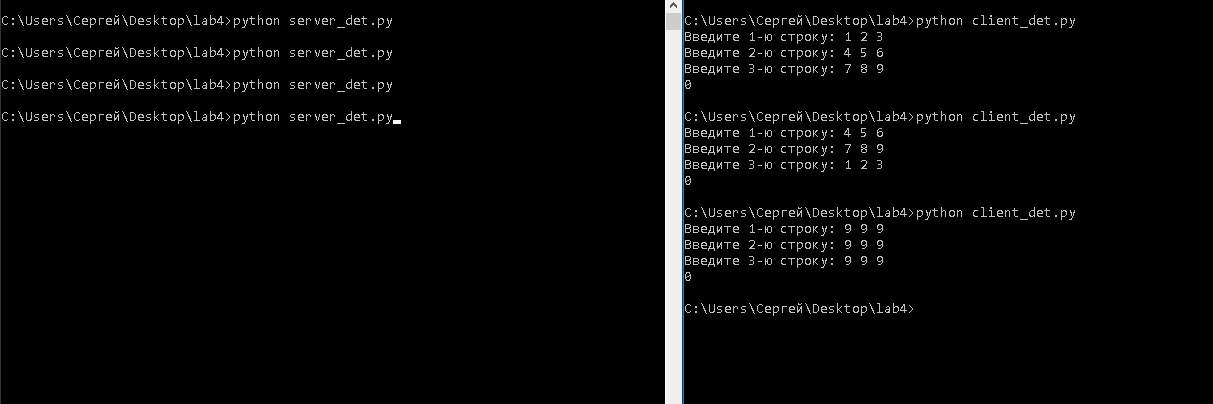


Рисунок 3 - Результат выполнения третей программы

* 1. Код клиента четвертого задания

import socket

sock = socket.socket()

sock.connect(('127.0.0.1', 9090))

while True:

user\_message = input('Вы: ')

sock.send(user\_message.encode()

if user\_message.upper() == "STOP":

break

answer = sock.recv(1024).decode()

print("Сервер: ", answer)

sock.close()

* 1. Код сервера четвертого задания

import socket

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

last\_message = 'Поступившего ранее сообщения не было'

while True:

message = conn.recv(1024).decode()

if message.upper() == "STOP":

sock.close()

break

conn.send(last\_message.encode())

last\_message = message

* 1. Скриншот выполнения программ четвертого задания

Скриншот выполнения представлен на рисунке 4

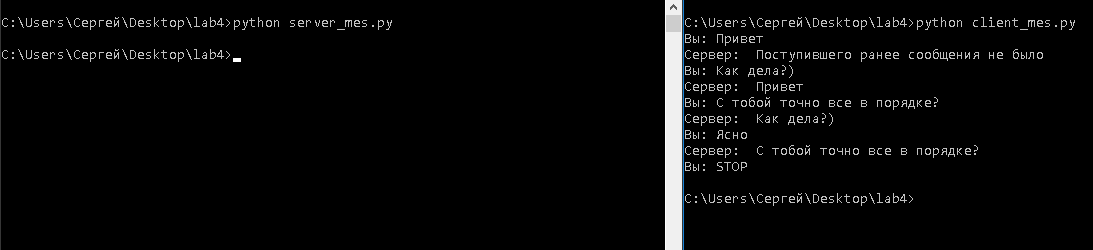


Рисунок 4 - скриншот выполнения программ 4 задания

* 1. Код клиента пятого задания

import socket

sock = socket.socket()

sock.connect(('127.0.0.1', 9090))

user\_message = ''

for i in range(0,2):

user\_message += input(f'Введите {i+1}-ю строку: ') + '

sock.send(user\_message.encode())

print(sock.recv(1024).decode())

* 1. Код сервера пятого задания

import socket

def det(data):

return data[0] \* data[3] - data[1] \* data[2]

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

data = conn.recv(1024).decode().strip().split(' ')

data = [int(x) for x in data]

data = [x \* det(data) for x in data]

conn.send(str(data).encode())

sock.close()

* 1. Скриншот выполнения программ пятого задания

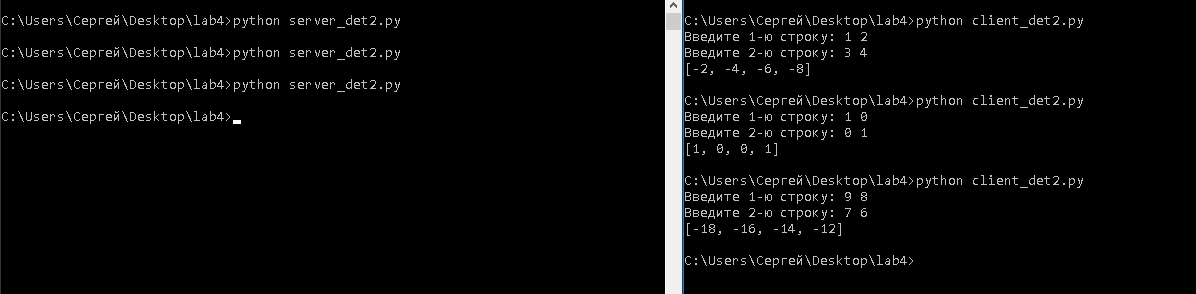


Рисунок 5 - Результат выполнения программ пятого задания

4. Ссылка на репозиторий: <https://github.com/psyduck31/master>

5. Выполнив данную лабораторную работу, я познакомился с клиент-серверными приложениями в языке программирования Python по средствам библиотеки socket.