**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 №4**

по дисциплине «Теория языков программирования»

Выполнил:

студент группы БСТ1801

Трибрат Дмитрий Олегович

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.

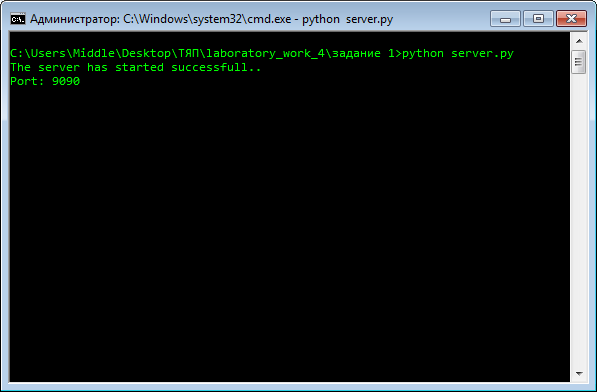
* 1. Четвертое задание:

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. Скриншот интерфейса сервера в любом приложении:



* 1. Код приложения клиента первой программы

import socket

sock = socket.socket()

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

message = input('Хотите ли вы узнать время? Y/N: ')

sock.send(message.encode())

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

while True:

message = input('Хотите ли вы узнать время? Y/N: ')

if message.lower() == "y":

sock.send(message.encode())

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

elif message.lower() == "n":

sock.send(message.encode())

break

else:

continue

sock.close()

* 1. Код приложения сервера первой программы

from time import localtime, strftime

import socket

print("The server has started successfull..\nPort: 9090")

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

print('\nСonnected:', addr)

while True:

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

if data.lower() == "y":

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

elif data.lower() == "n":

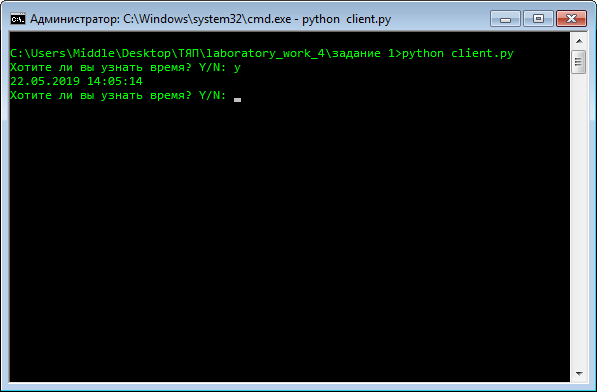
conn.close()

break

else:

continue

* 1. Скриншот выполнения программ первого задания



* 1. Код клиента второго задания

import socket

sock = socket.socket()

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

print("\nЧтобы закрыть программу, введите N: ")

message = input('Введите значения[a, b, c] через пробел: Y/N: ')

sock.send(message.encode())

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

while True:

message = input('\nВведите значения через пробел: ')

if message.lower() == "n":

sock.send(message.encode())

break

else:

sock.send(message.encode())

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

sock.close()

* 1. Код сервера второй программы

import socket

print("The server has started successfull..\nPort: 9090")

def func(data):

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

if c != a:

return abs(1 - a\*b\*\*c - a\*(b\*\*2 - c\*\*2) + (b - c + a) \* (12 + b)/(c-a))

else:

return "Деление на 0 невозможно!"

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

print('\nСonnected:', addr)

while True:

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

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

if data.lower() == "n":

conn.close()

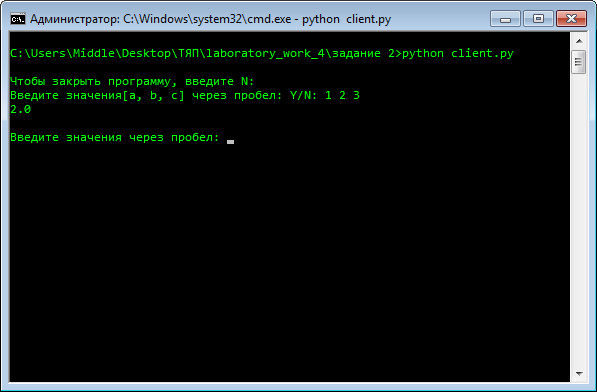
break

else:

answer = str(func(data))

conn.send(answer.encode())

* 1. Скриншот выполнения программ второго задания



* 1. Код клиента третьего задания

import socket

sock = socket.socket()

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

matrix = ''

for i in range(0,3):

matrix += input(f'Заполните {i+1}-ый ряд матрицы: ') + ' '

sock.send(matrix.encode())

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

while True:

for i in range(0,3):

matrix += input(f'Заполните {i+1}-ый ряд матрицы: ') + ' '

sock.send(matrix.encode())

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

* 1. Код сервера третьего задания

import socket

print("The server has started successfull..\nPort: 9090")

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()

print('\nСonnected:', addr)

while True:

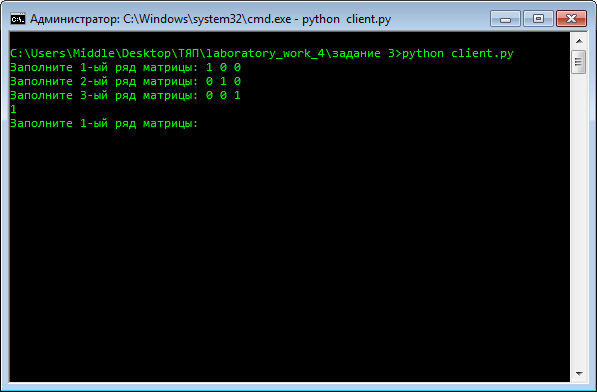
data = conn.recv(1024).decode().strip().split(' ')

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

answer = str(det(data))

conn.send(answer.encode())

* 1. Скриншот выполнения программ третьего задания



* 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.lower() == "n":

break

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

print("Сервер: ", answer)

sock.close()

* 1. Код сервера четвертого задания

import socket

print("The server has started successfull..\nPort: 9090")

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

last\_message = 'У вас нет сообщений...'

while True:

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

if message.lower() == "n":

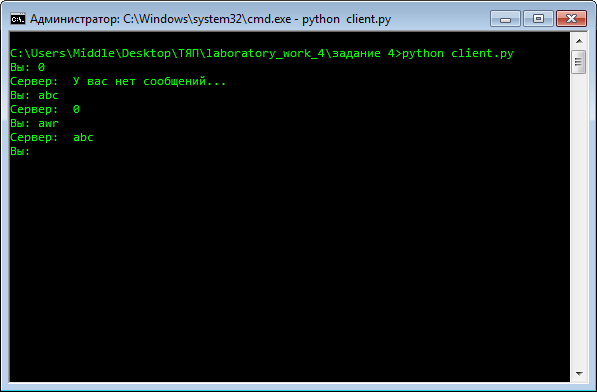
sock.close()

break

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

last\_message = message

* 1. Скриншот выполнения программ четвертого задания



* 1. Код клиента пятого задания

import socket

sock = socket.socket()

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

matrix = ''

for i in range(0,2):

matrix += input(f'Заполните {i+1}-ый ряд матрицы: ') + ' '

sock.send(matrix.encode())

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

while True:

for i in range(0,2):

matrix += input(f'Заполните {i+1}-ый ряд матрицы: ') + ' '

sock.send(matrix.encode())

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

* 1. Код сервера пятого задания

import socket

print("The server has started successfull..\nPort: 9090")

def det(data):

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

sock = socket.socket()

sock.bind(('', 9090))

sock.listen()

conn, addr = sock.accept()

print('\nСonnected:', addr)

while True:

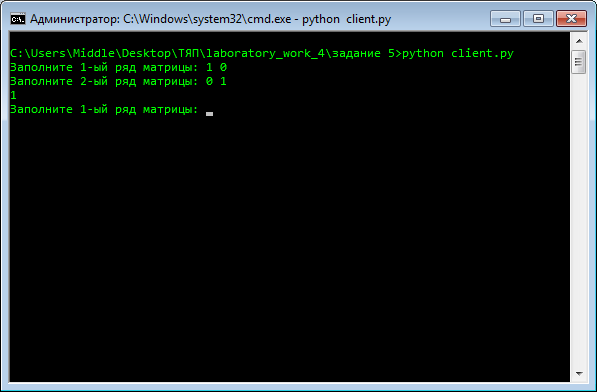
data = conn.recv(1024).decode().strip().split(' ')

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

answer = str(det(data))

conn.send(answer.encode())

* 1. Скриншот выполнения программ пятого задания



1. Ссылка на репозиторий: https://github.com/Dementr
2. Выполнив данную лабораторную работу, я познакомился с клиент-серверными приложениями в языке программирования Python по средствам библиотеки socket.