

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ОТЧЕТ**  
**по лабораторной работе**  
**по дисциплине «Функциональное программирование»**  
**на тему «Разработка асинхронного чат-сервера»**

Студент гр. 22.Б16-пу

\_\_\_\_\_

Горских Н.С

Преподаватель

\_\_\_\_\_

Киямов Ж.У.

**Санкт-Петербург**  
**2023 г.**

## **Оглавление**

<b>1. Цель работы</b>	<b>3</b>
<b>2. Задача</b>	<b>3</b>
<b>3. Описание алгоритма</b>	<b>3</b>
<b>4. Описание программы</b>	<b>3</b>
<b>4.1. Описание функций</b>	<b>4</b>
<b>5. Рекомендации пользователя</b>	<b>5</b>
<b>6. Рекомендации программиста</b>	<b>5</b>
<b>7. Вывод</b>	<b>5</b>
<b>8. Библиография</b>	<b>6</b>

## **1. Цель работы**

Создание чат-приложения с использованием асинхронного программирования и графического интерфейса для клиента.

## **2. Задача**

Разработать клиент-серверное чат-приложение, где клиенты могут подключаться к серверу, выбирать уникальные имена, общаться в общем чате и создавать/покидать персональные комнаты.

## **3. Описание алгоритма**

Клиент подключается к серверу, выбирает уникальное имя, после чего может отправлять сообщения в общий чат или создавать/покидать комнаты. Сервер управляет соединениями клиентов, обрабатывает сообщения и управляет комнатами.

## **4. Описание программы**

Программа представляет собой клиент-серверное чат-приложение с графическим интерфейсом на основе библиотеки Tkinter.

### **4.1. Описание функций**

Имя функции	Описание функции
<code>tk_main(root):</code>	Асинхронная функция обновления графического интерфейса Tkinter.
<code>send_message(writer, message)</code>	Асинхронная функция отправки сообщения серверу.

receive_messages(reader)	Асинхронная функция приема сообщений от сервера.

Таблица 5.1. Описание функций

get_username(reader, writer)	Асинхронная функция получения имени пользователя.
main(root)	Асинхронная функция основного цикла клиента, включающая ввод имени и обработку сообщений.
handle_client(reader, writer)	Асинхронная функция обработки подключения клиента на сервере.
handle_message(username, message, writer)	Асинхронная функция обработки сообщений от клиента.
join_room(username, room_name, writer)	Асинхронная функция присоединения клиента к комнате.
leave_room(username, room_name, writer)	Асинхронная функция выхода клиента из комнаты.
broadcast(message, sender)	Асинхронная функция рассылки сообщения всем клиентам.
send_message(writer, message)	Асинхронная функция отправки

	сообщения конкретному клиенту.
<code>start_server(host, port)</code>	Асинхронная функция запуска сервера на указанном хосте и порту.
<code>send_pm(username, message, writer)</code>	Асинхронная функция отправки личного сообщения по никнейму

## 5. Рекомендации пользователя

Запускайте сервер перед запуском клиентов.

При подключении укажите уникальное имя пользователя.

Используйте команды `/join room` и `/leave room` для управления комнатами.

Используйте команду `/pm` для отправки личного сообщения пользователю.

Для выхода из приложения введите `exit`.

## 6. Рекомендации программиста

Для запуска программы необходим Python3 и пакеты к нему:

`asyncion`, `aioconsole`, `tkinter`.

## 7. Вывод

Реализовано асинхронное клиент-серверное чат-приложение с графическим интерфейсом, позволяющее пользователям общаться в общем чате и управлять персональными комнатами.

## 8. Библиография

Документация asyncio: <https://docs.python.org/3/library/asyncio.html>

Документация tkinter: <https://docs.python.org/3/library/tkinter.html>

Репозиторий с кодом:

<https://github.com/peshkanavalnogo/asyncio-client-server-chat>