

Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Омский государственный аграрный университет им. П. А. Столыпина»

Университетский колледж агробизнеса

Инженерное отделение

Специальность 09.02.07 Информационные системы и программирование

**КУРСОВАЯ РАБОТА**  
**МДК 02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО**  
**ОБЕСПЕЧЕНИЯ**

**ТЕМА: Онлайн чат**

Исполнитель :  
Бондаренко Дмитрий  
Александрович  
Студент 2 курса группы 21 ИС  
Руководитель курсовой работы:  
Кортусов Алексей Владимирович  
Оценка: \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

г. Омск  
2023

## **Содержание**

Введение	3
Выбор среды программирования	4
Анализ аналогичных программ	6
Актуальность и область применения	8
Описание структуры программного продукта	9
Интерфейс программного приложения	10
Требования предъявляемые к системе для корректной работы	11
Тестирование программы	12
Заключение	13
Список используемых источников	14
Приложение А	16
Приложение В	17

## **Введение**

Создание программного средства - динамически длительный и трудоемкий процесс. Современные технологии проектирования основаны на последовательной (поэтапной) разработке. По общности целей последовательности работ (этапы) обычно объединяются в стадии.

Сейчас, в наше время, людям зачастую требуется возможность коммуницировать друг с другом в режиме реального времени, для этого были созданы мессенджеры, онлайн чаты.

Технология разработки программного обеспечения (ТРПО) - система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах.

Курсовая работа по дисциплине ТРПО посвящена разработке онлайн чата на основе универсального языка программирования Python используя библиотеку PyWebIO.

Цель курсовой работы - создать онлайн чат для общения в браузере.

Для достижения цели необходимо выполнить следующие задачи:

1. Знакомство с библиотекой PyWebIO.
2. Создание онлайн чата.
3. Тестирование программного продукта.

## **Выбор среды программирования**

Для разработки программного продукта был выбран язык программирования Python. Python - это высокоуровневый язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении. И библиотека PyWebIO - это разнообразный набор функций обязательных для получения пользовательского ввода и вывода содержимого в браузере, превращая браузер в "терминал форматированного текста", и может использоваться для создания простых веб-приложений или приложений с графическим интерфейсом на основе браузера.

### **Python обладает такими преимуществами:**

Лёгкость освоения - Python прост, логичен, имеет понятный синтаксис, поэтому подходит даже для новичков. Для его изучения не нужны знания английского.

Простота визуального восприятия - блоки кода отделяются отступами, за счёт меньшего объёма код также воспринимается проще.

Кроссплатформенность - программы, написанные на Python, могут запускаться и функционировать на всех типах операционных систем. Отличия можно узнать заранее, поскольку они подробно описаны в документации.

Скорость разработки - чтобы написать программу на Python нужно значительно меньше кода, чем при разработке, например - на Java.

Универсальность - благодаря интерпретируемости python он используется практически на всех платформах для различных задач - от тестирования до научных исследований. Код удобно писать даже в стандартных текстовых редакторах.

Множество инструментов - стандартные библиотеки Python способны решать даже сложные задачи. Установка дополнительных модулей, созданных для конкретных целей, помогает при разработке специальных проектов.

Масштабируемость - возможность адаптации высокоуровневой логики позволяет проектам, разработанным на Python, масштабироваться и расширяться.

### **Python обладает такими недостатками:**

Медленная работа - этот недостаток лишает пайтон-разработчиков возможности создавать высокопроизводительные проекты только на Python. Необходимо задействовать другие языки программирования. Программное обеспечение работает медленно из-за устаревшего кода Global Interpreter Lock, сохранившимся её с первой версии языка.

Трудность переноса проектов на другие системы - проблема возникает из-за зависимости языка программирования от библиотек.

Ресурсоёмкость - Python не подходит для проектов, требующих больших объёмов памяти.

Все недостатки Python можно аннулировать, используя дополнительные библиотеки, код внутри которых написан на быстрых низкоуровневых языках. Сочетая удобство, читаемость python и скорость работы, к примеру C++, мы получаем высококачественный проект.

## Анализ аналогичных программ

1. ICQ - бесплатная кроссплатформенная система мгновенного обмена сообщениями, для мобильных и иных платформ с поддержкой голосовой и видеосвязи. Позволяет пересылать текстовые сообщения, изображения, видео и аудио через Интернет.

Служба является коммерческой, но её использование бесплатно. С момента создания служба принадлежала своему разработчику, израильской компании Mirabilis, в 1998 году она была продана американской компании AOL, а в апреле 2010 года — российскому инвестиционному фонду Digital Sky Technologies. После реорганизации DST в сентябре 2010 года ICQ вошла в состав Mail.ru Group (ныне VK).

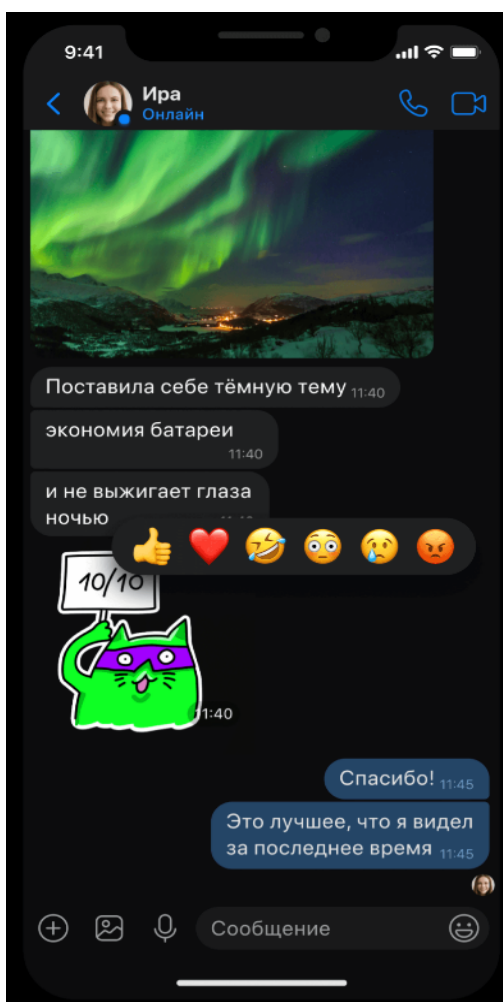


Рис 1. "ICQ"

2. Telegram - кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями обмена текстовыми, голосовыми и видео сообщениями, а также стикерами, фотографиями и файлами многих форматов. Также можно совершать аудио и видеозвонки, устраивать трансляции в каналах и группах, организовывать конференции, многопользовательские группы и каналы. Проект создан Павлом Дуровым, основателем социальной сети «ВКонтакте».

В интервью The New York Times Павел рассказал, что первоначальная идея приложения пришла ему ещё в 2011-м, когда к его двери приходили спецназовцы. Когда последние всё-таки ушли, Дуров сразу же написал своему брату Николаю. Тогда же он и осознал, что у него нет безопасного способа коммуникации с братом. Сервис построен на технологии шифрования переписки MTProto, разработанной братом Павла — Николаем. Сам «Телеграм» изначально был экспериментом принадлежащей Павлу компании Digital Fortress с целью протестировать MTProto на больших нагрузках.

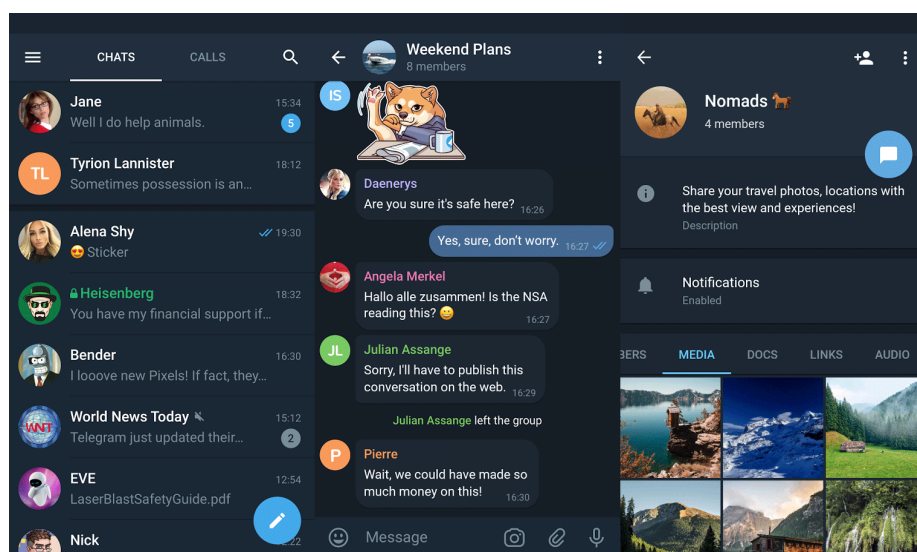


Рис 2. "Telegram"



## **Актуальность и область применения**

Система общественных отношений тесным образом связана с развитием информационного пространства, а ее формирование и развитие напрямую зависит от изменений, происходящих во всех сферах жизни.

Характерное для современного общества широкое распространение информационных технологий, привели к появлению новых форм общения, особого вида коммуникативного взаимодействия между людьми, получившего в науке название интернет-коммуникации.

Все эти нововведения позволяют облегчить нам жизнь в каких-либо моментах.

Этот проект упростит связь людей между собой (к примеру в кабинете) , ведь он позволит осуществлять общение не сдвигаясь с места, и не отвлекая никого.

## Описание структуры программного продукта

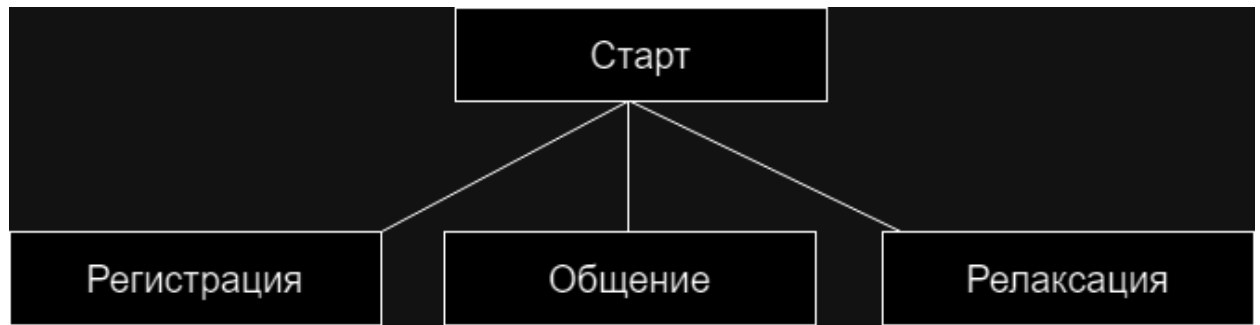


Схема 1. Структурная схема

Онлайн чат состоит из нескольких модулей:

### 1.Главный модуль

-Инициализирует пользовательский интерфейс на основе библиотеки PyWebIO.

### 2. Модуль взаимодействия с интерфейсом.

- Содержит форму регистрации, просмотр и отправку сообщений.

### 3. Модуль сервера.

- Запускает сервер на локальной сети.

## Интерфейс программного приложения

Интерфейс – это набор инструментов, который позволяет пользователю взаимодействовать с программой. Под интерфейсом понимается набор средств, правил и методов, за счет которых осуществляется коммуникация между элементами системы, различными программами и устройствами. На основе интерфейса реализуется взаимодействие всех современных информационных систем.

PyWebIO – это простая библиотека на языке python. Она используется для создания простых веб-приложений или приложений с графическим интерфейсом на основе браузера. Она не требует знаний HTML и JS. PyWebIO идеально подходит для быстрого создания интерактивных приложений, не требующих сложного пользовательского интерфейса.

В данном проекте реализован простой пользовательский интерфейс. Меню содержит кнопки первоначальной регистрации, а после кнопки ввода/отправки сообщений.

И выглядит он так:

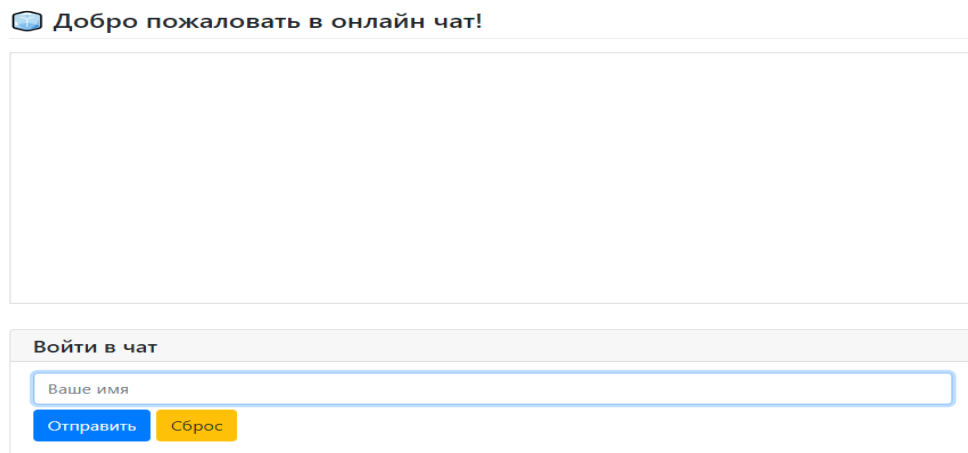


Рис 4. Онлайн чат.

## **Требования предъявляемые к системе для корректной работы**

### **Минимальные системные требования:**

Операционная система: Windows 7/8/10/11

Процессор: Intel core i5 2405s

Оперативная память: 4 Гб

Видеоадаптер: Nvidia GeForce 550

### **Рекомендуемые системные требования:**

Операционная система: 10/11

Процессор: Intel core i5 2500k

Оперативная память: 8 Гб

Видеоадаптер: Nvidia GeForce 550

## Тестирование программы.

Исходя из анализа видов тестирования, было выбрано тестирование в определенной среде, так в данном проекте не так много функций, так же нет входных параметров.

Таблица 1 — Тестирование

<b>Система</b>	<b>Разрешение экрана</b>	<b>Ошибки/несоответствия</b>
Windows 10	1920x1080	Нет
Windows 8	1280x1024	Нет
Windows 7	1024x960	Нет

## **Заключение**

В ходе выполнения текущей работы были повышены знания навыки программирования на языке Python. Поставленные задачи и цели были достигнуты. В результате выполнения курсовой работы был полностью реализован “Онлайн чат”. В ходе тестирования, ошибок не обнаружено.

Были проанализированы популярные средства разработки. В ходе анализа, было проведено их сравнение и выбраны наиболее актуальные средства разработки для начинающих разработчиков. Выбор приоритетных средств разработки проходил по трём критериям: доступность, функциональность и понятность.

При анализе существующих разработок, был проведено их сравнение и выделены их достоинства и недостатки. В ходе анализа стало ясно, что при разработке “Онлайн чат” с простым функционалом, стоит обратить внимание на дополнительные элементы, такие как: удобство интерфейса, функциональность. Это нужно для того, чтобы привлечь новых пользователей и продлить жизненный цикл разработанного программного приложения. Такие образом поставленные задачи выполнены и цель достигнута.

## **Список используемых источников**

1. Банкрашков, А.В. Программирование для детей на языке Python / А.В. Банкрашков. - М.: АСТ, 2018. - 288 с.
2. Прохоренок, Николай Python 3 и PyQt. Разработка приложений / Николай Прохоренок. - М.: БХВ-Петербург, 2020. - 704 с.
3. Абрахам, Никхил Веб-программирование для чайников / Никхил Абрахам. - М.: Вильямс, 2020. - 304 с.
4. Матиз Эрик. Изучаем python: программирование игр, визуализация данных, веб-приложения. 3-е издание - СПб.: Питер, 2021 - 512 с.
5. Свейгарт Эл. Учим python, делая крутые игры / Эл Свейгарт ; [Перевод с английского М.А.Райтмана] - Москва: Эксмо, 2018 - 416 с.
6. Макманус шон. Миссия python создаем игры вместе с детьми / Шок макманус ; [Перевод с английского С.В. Черникова] - Москва: Эксмо, 2022 - 384
7. Яворски Михал, Зиаде Тарек. Python. Лучшие практики и инструменты. - СПб.: Питер, 2021 - 560 с.
8. Дронов, Владимир Django: практика создания Web-сайтов на Python / Владимир Дронов. - М.: БХВ-Петербург, 2020. - 176 с.
9. Любанович, Билл Простой Python. Современный стиль программирования / Билл Любанович. - М.: Питер, 2023. - 480 с.

10. Пэйн, Брайсон. Python для детей и родителей / Брайсон Пэйн ; перевод с английского М.А. Райтман]. - 2-е издание. - Москва : Эксмо, 2021 -352 с.
11. <https://metanit.com/python/tutorial/> (дата обращения: 04.12.2023)
12. <https://code-basics.com/ru/languages/python> (дата обращения: 04.12.2023)
13. <https://itproger.com/course/python> (дата обращения: 01.12.2023)
14. <https://pythonworld.ru/> (дата обращения: 26.11.2023)
15. PyWebIO: <https://pywebio.readthedocs.io> (дата обращения: 22.11.2023)
16. <http://server.aesc.msu.ru/materials/PYTHON/pythonworldru.pdf>(дата обращения 25.11.2023)
17. [https://stepik.org/course/67/promo?utm\\_source=feed\\_april&utm\\_medium=organic&utm\\_campaign=67](https://stepik.org/course/67/promo?utm_source=feed_april&utm_medium=organic&utm_campaign=67) (дата обращения 15.12.2023)
18. <https://wiki.python.org/moin/BeginnersGuide>(дата обращения 27.11.2023)
19. <https://devguide.python.org> (дата обращения 27.11.2023)
20. <https://www.youtube.com/watch?v=cfJrtx-k96U> (дата обращения 20.11.2023)



## Приложение А



## Приложение В

```

import asyncio

from pywebio import start_server
from pywebio.input import *
from pywebio.output import *
from pywebio.session import defer_call, info as session_info, run_async,
run_js

chat_msgs = []
online_users = set()

MAX_MESSAGES_COUNT = 100

async def main():
    global chat_msgs

    put_markdown("## Добро пожаловать в онлайн чат!")

    msg_box = output()
    put_scrollable(msg_box, height=300, keep_bottom=True)

    nickname = await input("Войти в чат", required=True, placeholder="Ваше имя", validate=lambda n: "Такой ник уже используется!" if n in online_users or n == '' else None)
    online_users.add(nickname)

    chat_msgs.append((' ', f'`{nickname}` присоединился к чату!'))
    msg_box.append(put_markdown(f'`{nickname}` присоединился к чату'))

    refresh_task = run_async(refresh_msg(nickname, msg_box))

    while True:
        data = await input_group(" Новое сообщение", [
            input(placeholder="Текст сообщения ...", name="msg"),
            actions(name="cmd", buttons=["Отправить", {'label': "Выйти из чата", 'type': 'cancel'}])
        ], validate = lambda m: ('msg', "Введите текст сообщения!") if m["cmd"] == "Отправить" and not m['msg'] else None)

        if data is None:
            break

        msg_box.append(put_markdown(f"``{nickname}`: {data['msg']}"))
        chat_msgs.append((nickname, data['msg']))

    refresh_task.close()

    online_users.remove(nickname)
    toast("Вы вышли из чата!")
    msg_box.append(put_markdown(f' Пользователь `{nickname}` покинул чат!'))
    chat_msgs.append((' ', f'Пользователь `{nickname}` покинул чат!'))

```

```

        put_buttons(['Перезайти'], onclick=lambda
btn:run_js('window.location.reload()'))

async def refresh_msg(nickname, msg_box):
    global chat_msgs
    last_idx = len(chat_msgs)

    while True:
        await asyncio.sleep(1)

        for m in chat_msgs[last_idx:]:
            if m[0] != nickname: # if not a message from current user
                msg_box.append(put_markdown(f"`{m[0]}`: {m[1]}"))

        # remove expired
        if len(chat_msgs) > MAX_MESSAGES_COUNT:
            chat_msgs = chat_msgs[len(chat_msgs) // 2:]

        last_idx = len(chat_msgs)

if __name__ == "__main__":
    start_server(main, debug=True, port=8080, cdn=False)

```