

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»  
(СибГУТИ)

09.03.01 Информатика и вычислительная  
техника

Профиль: Программное обеспечение  
средств вычислительной техники и авто-  
матизированных систем  
(очная форма обучения)

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ  
на Кафедре вычислительных систем

Разработка Telegram-бота для поиска информации об известных  
людях

Выполнил:

студент института ИВТ

гр. ИП-015

«25» мая 2023г.

\_\_\_\_\_/Шевельков П.С./

(подпись)

Проверил:

Руководитель от СибГУТИ

«25» мая 2023г.

\_\_\_\_\_/Приставка П.А./

(подпись)

Новосибирск 2023

**План-график проведения \_\_\_\_\_ производственной \_\_\_\_\_ практики**

Вид практики

Шевелькова Павла Сергеевича

Фамилия Имя Отчество студента

института Информатика и вычислительная техника, 3 курса, гр. ИП-015

Направление: 09.03.01 – Информатика и вычислительная техника

Код – Наименование направления (специальности)

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

Место прохождения практики Кафедра вычислительных систем

Объем практики: **360/10** часов/ЗЕ

Вид практики ***производственная***

Тип практики ***Технологическая (проектно-технологическая) практика***

Срок практики с "30" января 2023 г.  
по "27" мая 2023 г.

Содержание практики\*:

Наименование видов деятельности	Дата (начало – окончание)
1. Общее ознакомление со структурным подразделением предприятия, вводный инструктаж по технике безопасности	30.01.2023–02.02.2023
2. Выдача задания на практику, деление студентов на группы (если необходимо), определение конкретной индивидуальной темы, формирование плана работ	03.02.2023–05.02.2023
3. Работа с библиотечными фондами структурного подразделения или предприятия, сбор и анализ материалов по теме практики	07.02.2023–12.02.2023
4. Выполнение работ в соответствии с составленным планом: – Определение технологий, с которыми будет вестись работа – Изучение информации об API Telegram, в частности библиотеки aiogram и других – Создание бота – Формирование сообщения и его отправка	14.02.2023 – 21.05.2023
5. Анализ полученных результатов и произведенной работы Составление отчета по практике, защита отчета	23.05.2023–27.05.2023

\*В соответствии с программой практики

Руководитель от СибГУТИ

\_\_\_\_\_/\_\_\_\_\_/

«        »

2023г.

(подпись)

## **ЗАДАНИЕ НА ПРАКТИКУ**

Разработка Telegram-бота для поиска биографии известных личностей.

Работу можно разбить на несколько этапов:

- 1) Определение технологий, с которыми будет вестись работа.
- 2) Изучение информации об API Telegram, в частности библиотеки aiogram и других.
- 3) Создание бота
- 4) Формирование сообщения и его отправка.

## ВВЕДЕНИЕ

С каждым днём люди придумывают всё больше новых способов/приложений для удобства пользования в сфере интернета. Так вот одно из приложений - Telegram, которое собрало в себе множество удобных и нужных функций. Рассмотрим одну из функций - бот. Что это такое и для чего?

Бот — это небольшое приложение, которое самостоятельно выполняет заранее созданные задачи без участия пользователя. Telegram-бот умеет делать всё, что мог бы делать человек в чате: отвечать на вопросы, присылать ссылки на сайты или создавать мемы. Автоматически или по запросу он может отправлять:

- текстовые сообщения;
- картинки;
- видео;
- файлы.

Боты умеют:

- Выполнять действия, которые нельзя настроить на канале. Например, продавать товары и принимать оплату, общаться с пользователями, скрывать личные данные. Боты для Телеграма могут собирать потенциальных клиентов, подключая их к CRM, системе продажи билетов или платформе обмена сообщениями.
- Выполнять несколько разных команд одновременно. Важная функция ботов — возможность запускать цепочку действий, постепенно запрашивая у пользователя новую информацию. Если отправить боту команду /start, на экране появятся кнопки.
- Интегрироваться с другими сервисами. Например, можно настроить бота, чтобы пользователи могли выполнять быстрый поиск GIF-анимации во встроенном режиме. Если встроенные запросы будут включены, пользователи смогут ввести имя бота и свой запрос в поле ввода текста в любом чате и мгновенно получить нужную гифку.

## **ПОСТАНОВКА ЗАДАЧИ**

Для реализации бота для перевода слов потребуется получить источник откуда будем брать перевод. Перевод буду брать с сайта wikipedia. Для того, чтобы взять с сайта данные, нам потребуется подключить соответствующую библиотеку. Задача – разобраться с тем, как правильно использовать библиотеку, и как запрограммировать нашего бота, по требованию пользователя, эту информацию предоставить.

## ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

В качестве языка программирования был выбран Python версии (3.11). Он располагает всем необходимым инструментарием для написания данной программы.

**Python** — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++.

Python является мультипарадигменным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование. Задачи обобщённого программирования решаются за счёт динамической типизации. Аспектно-ориентированное программирование частично поддерживается через декораторы, более полноценная поддержка обеспечивается дополнительными фреймворками. Такие методики как контрактное и логическое программирование можно реализовать с помощью библиотек или расширений. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Стандартная библиотека включает большой набор полезных переносимых функций, начиная с возможностей для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на Си или C++, при этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках. Существует и специализированный репозиторий программного обеспечения, написанного на Python, — PyPI. Данный репозиторий предоставляет средства для простой установки пакетов в операционную систему и стал стандартом де-факто для Python. По состоянию на 2019 год в нём содержалось более 175 тысяч пакетов.

В качестве среды для программирования была выбрана IDE PyCharm от JetBrains, так как она предоставляет удобный инструментарий для работы с данным языком, такой как: удобная установка библиотек, подсветка синтаксиса, рекомендации по написанию кода и т.д.

**PyCharm** — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик. Для создания бота использовалась библиотека aiogram.

**Aiogram** - это асинхронный python фреймворк для создания telegram-ботов.

## ВЫПОЛНЕНИЕ РАБОТЫ

**Этап 1:** Определение технологий, с которыми будет вестись работа

Для написания Telegram ботов существует удобный инструмент на языке Python. Из библиотек для создания бота был выбран aiogram, так как он представляет из себя простое и понятное решение, полностью покрывающее наши запросы.

**Этап 2:** Изучение информации об API Telegram, в частности библиотеки aiogram и других.

Были прочитаны документация и статьи на тему библиотек aiogram, Bot, Dispatcher, executor, types.

**Этап 3:** Создание бота.

Прежде чем начинать разработку, бота необходимо зарегистрировать и получить его уникальный id, являющийся одновременно и токеном. Для этого в Telegram существует специальный бот — @BotFather.

Через него задаются внешние параметры бота такие как имя, главная иконка и т.д.

Токен который мы получили, в дальнейшем будем использовать для подключения к боту через Python.

```
token = '5833135411:AAELIPPiGIm2nh-Eot-iDaxw4tu-vMfbeDY'
```

```
bot = Bot(token=token)
```

**Этап 4:** Формирование сообщения и его отправка.

Ниже описаны функции для отправки сообщений на определенные команды, введенные пользователем.

```
@dp.message_handler(commands="Start")
```

```
async def start(message: types.Message):
```

```
    await message.answer("Здравствуйте, введите знаменитость, про которую хочется узнать побольше!")
```

Следующий handler обрабатывает любое сообщение от пользователя (кроме приведенного выше), и запускает весь алгоритм поиска личности с обработчиком запроса.

```
@dp.message_handler()
```

```
async def search(message: types.Message):
```

```
    query = message.text.strip()
```

```
    conn = sqlite3.connect('/home/pavlo/6 cernak/practic/bio.db')
```

```
    cursor = conn.cursor()
```

```
    dop_query = query.lower().strip().replace(' ', '')
```

```
    cursor.execute(f"SELECT * FROM name_bio WHERE FIO = ?", (f'{dop_query}',))
```

```
    result = cursor.fetchone()
```

```
    if result:
```

```
        await bot.send_message(message.chat.id, text="Недопустимый запрос")
```

```
        await send_notification(query)
```

```
        return
```

```
    url = f"https://ru.wikipedia.org/wiki/{query}"
```

```
    soup = make_request(url)
```

```
    image_person = find_image(soup)
```

```
    info = parsing_person(soup)
```

```
    if image_person:
```

```
        await bot.send_photo(message.chat.id, photo=image_person, caption=info)
```

```
    else:
```

```
        await message.reply(info)
```

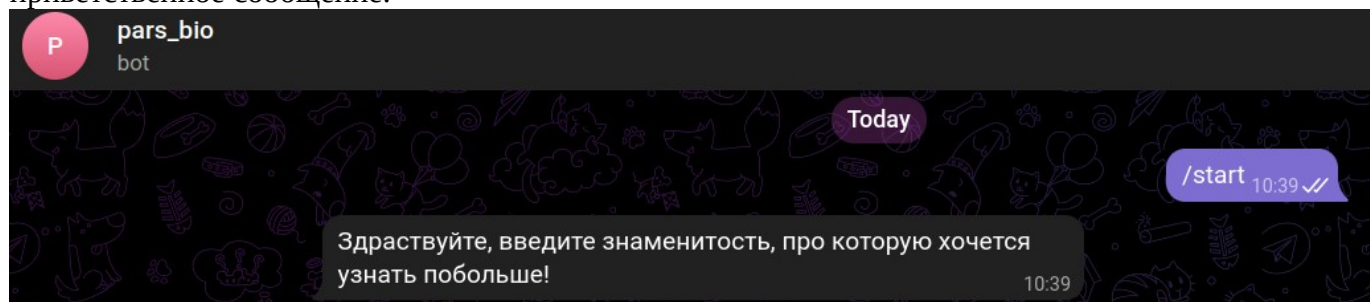
Функция send\_notification отправляет уведомление создателю по id.

```
async def send_notification(query):
```

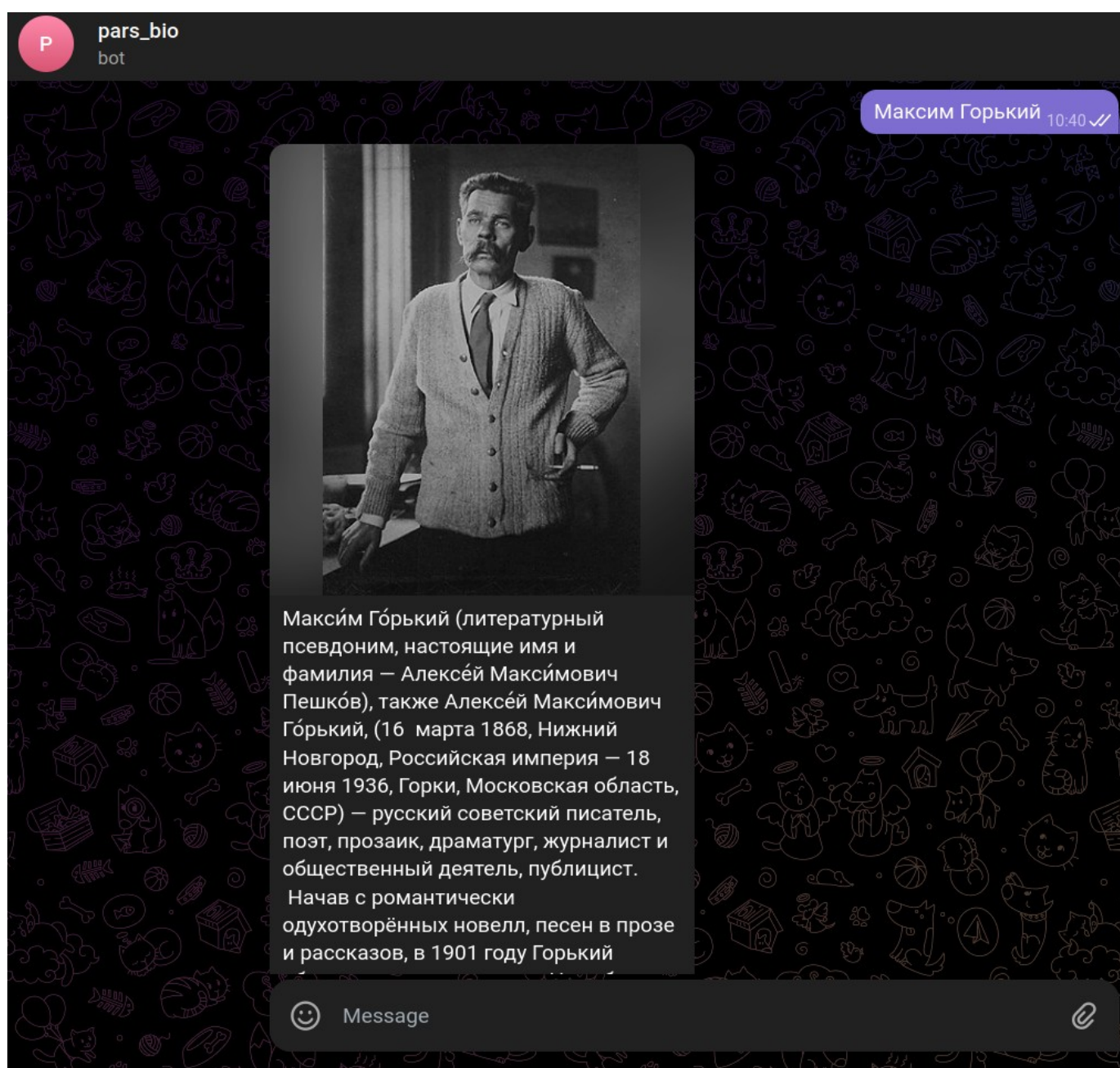
```
    await bot.send_message(chat_id='1381925', text=(f'Искали нехорошее - {query}'),  
    disable_notification=True)
```

## ОПИСАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

При первом запуске пользователя будет ожидать кнопка START, где, нажав на неё появится приветственное сообщение.

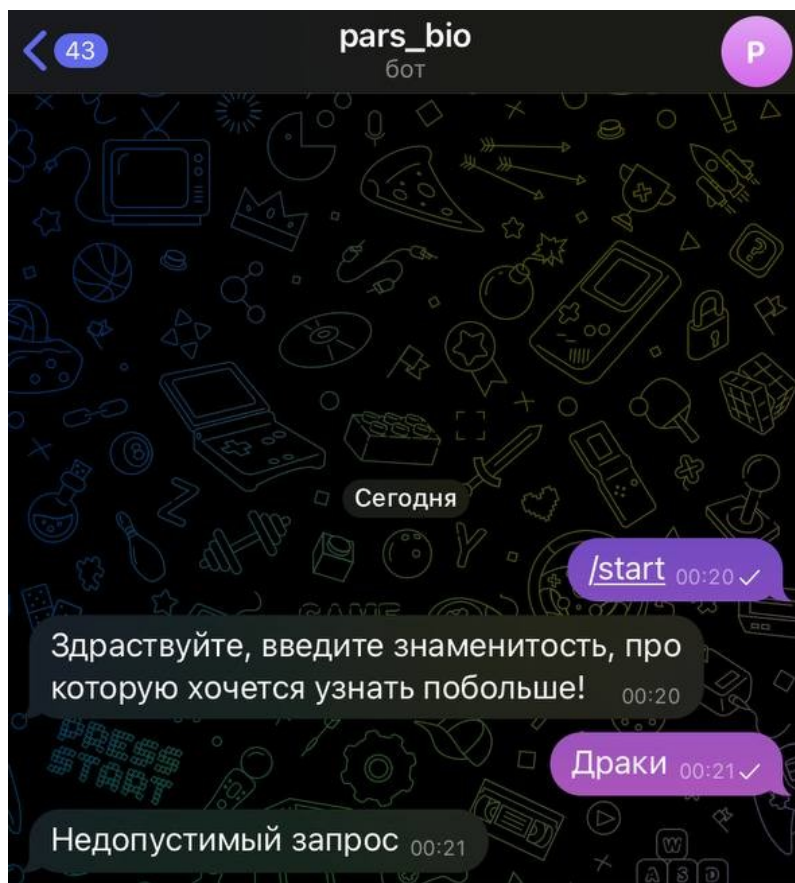


Далее пользователь уже знает, что нужно делать, поэтому вводит имя и фамилию известного человека, о котором ему нужна информация.

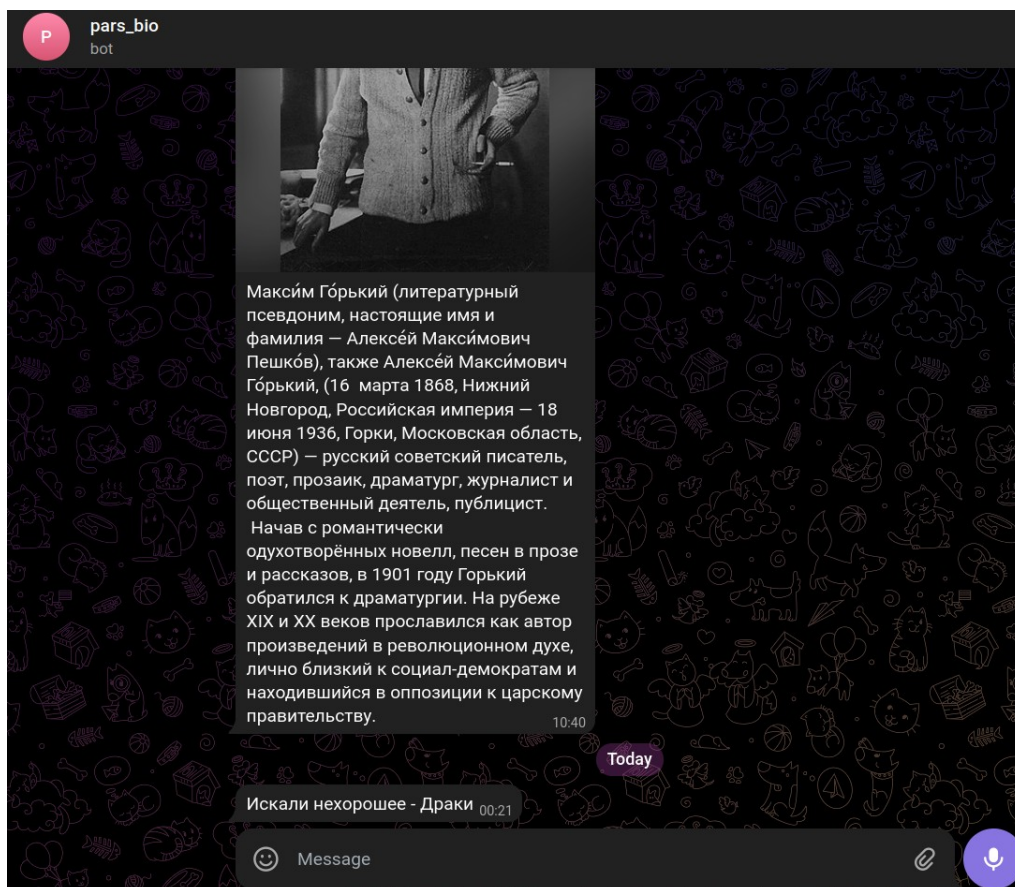




В случае «запрещённого запроса» пользователь получит соответствующие сообщение.



Создатель в этот момент получит уведомление о данном запросе.



## **ЗАКЛЮЧЕНИЕ**

Был создан бот для удобного и быстрого поиска информации об известных людях. Также в боте присутствует обработчик запросов, в случае «запрещённого запроса» создателю — в моём лице, приходит уведомление об данном запросе, а пользователю сообщение о «запрещённом запросе». В дальнейшем можно доработать функционал бота, добавив, например, поиск по запросу на английском языке. Сейчас бот выполняет базовую функцию - поиск информации об известных личностях, введённых пользователем, и со своей задачей справляется.

Получил опыт работы с различными библиотеками для создания ботов и работы с ними на языке Python.

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Ю.М. Шыырап, Л.Ю. Забелин, М.Ю. Шыырап Основы программирования Python: Учебное пособие / Сибирский государственный университет телекоммуникаций и информатики; каф. систем автоматизированного проектирования. – Новосибирск, 2017. – 197 с. №164 от 20.12.18.(Дата обращения 14.04.2023)
2. Документация по pyTelegramBotAPI [Электронный ресурс]: PyTelegramBot.URL: <https://pytba.readthedocs.io/ru/latest/index.html> (Дата обращения 15.04.2023)

## ПРИЛОЖЕНИЯ

### **tg\_bot.py:**

```
from aiogram import Bot, Dispatcher, executor, types
from pars_bio import *
import sqlite3
token = '5833135411:AAELlPPiGIm2nh-Eot-iDaxw4tu-vMfbeDY'
bot = Bot(token=token)
dp = Dispatcher(bot)
conn = sqlite3.connect('/home/pavlo/6 cemark/practic/bio.db')
@dp.message_handler(commands="Start")
async def start(message: types.Message):
    await message.answer("Здравствуйте, введите знаменитость, про которую хочется узнать побольше!")
@dp.message_handler()
async def search(message: types.Message):
    query = message.text.strip()
    conn = sqlite3.connect('/home/pavlo/6 cemark/practic/bio.db')
    cursor = conn.cursor()
    dop_query = query.lower().strip().replace(' ', '')
    cursor.execute(f"SELECT * FROM name_bio WHERE FIO = ?", (f'{dop_query}',))
    result = cursor.fetchone()
    if result:
        await bot.send_message(message.chat.id, text="Недопустимый запрос")
        await send_notification(query)
        return
    url = f"https://ru.wikipedia.org/wiki/{query}"
    soup = make_request(url)
    image_person = find_image(soup)
    info = parsing_person(soup)
    if image_person:
        await bot.send_photo(message.chat.id, photo=image_person, caption=info)
    else:
        await message.reply(info)
async def send_notification(query):
    await bot.send_message(chat_id='1381925', text=(f'Искали нехорошее - {query}'), disable_notification=True)
if __name__ == "__main__":
    executor.start_polling(dp)
```

### **pars\_bio.py:**

```
import requests
from bs4 import BeautifulSoup
def make_request(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    return soup
def parsing_person(soup):
    try:
        content_div = soup.find('div', {'id': 'mw-content-text'})
        paragraphs = content_div.find_all('p')
        info = ''.join([p.get_text() for p in paragraphs[:3]])
        info_red = ""
        i = 0
        while i < len(info):
            if info[i] == '[':
                while info[i] != ']':
                    i+=1
                i+=1
            else:
                info_red+=info[i]
                i+=1
        except Exception:
            print("Ошибка")
            info_red = "Не удалось найти информацию"
        finally:
            return info_red
def find_image(soup):
    try:
        image = soup.find('table').find('img')
        image = "https:" + image.get('src')
    except:
        image = None
    finally:
        return image
```

(ФИО студента)

(ФИО студента)

## Уровень освоения компетенций

Шевельков Павел Сергеевич

(ФИО студента)

Компетенции	Уровень сформированности компетенций
<i>ПК-1 - Способен разрабатывать требования и проектировать программное обеспечение</i>	

отметка о зачете \_\_\_\_\_

Руководитель практики от СибГУТИ:

\_\_\_\_\_  
Должность руководителя

\_\_\_\_\_  
подпись

\_\_\_\_\_  
ФИО руководителя

"\_\_" \_\_\_\_\_ 20\_\_ г