

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по ДЗ

Выполнил:

Студент группы ИУ5-33Б  
Абрамов А. В.

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Москва, 2023 г.

## Постановка задачи

Разработать программу на языке программирования Python, реализующую обращение к OpenAI API внутри Telegram бота через прокси.

## Текст программы

```
from openai import OpenAI
import httpx
import os
import asyncio
import logging
import sys

from aiogram import Bot, Dispatcher, types
from aiogram.filters import CommandStart
from aiogram.types import Message
from aiogram.enums import ParseMode

API_KEY = os.environ.get('API_KEY')
client = OpenAI(
    http_client=httpx.Client(
        proxies="http://user131354:g5o86q@85.209.107.210:1308"
    ),
    api_key=API_KEY
)

BOT_TOKEN = os.environ.get('BOT_TOKEN')
dp = Dispatcher()

@dp.message(CommandStart())
async def command_start_handler(message: Message) -> None:
    await message.answer("Этот бот позволяет общаться с ChatGTP")

@dp.message()
async def gpt_handler(message: types.Message) -> None:
    user_question = message.text

    try:
        response = client.chat.completions.create(
            messages=[
                {
                    "role": "user",
                    "content": user_question,
                }
            ],
            model="gpt-3.5-turbo",
        )

        # Извлекаем ответ от OpenAI
        bot_response = response.choices[0].message.content

        # Отправляем ответ пользователю
        await message.answer(bot_response, parse_mode=ParseMode.MARKDOWN)

    except Exception as e:
        logging.error(f"Error interacting with OpenAI API: {e}")
        await message.answer("Произошла ошибка при обработке запроса. Попробуйте еще раз позже.")

async def main() -> None:
```

```
bot = Bot(BOT_TOKEN, parse_mode=ParseMode.HTML)
await dp.start_polling(bot)

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO, stream=sys.stdout)
    asyncio.run(main())
```

## Результат выполнения программы

