

# ЛЕБЕДЕВ ОЛЕГ

[oglebedev222@gmail.com](mailto:oglebedev222@gmail.com) | [github.com/LebedevOleg1](https://github.com/LebedevOleg1)

## ОБРАЗОВАНИЕ

### НИЯУ МИФИ

Прикладная математика и информатика. Институт лазерных и плазменных технологий

Москва

3 курс

## ПРОЕКТЫ

### Library Telegram Bot | *Python, Telebot, cv2, pyzbar, PostgreSQL*

- На базе API telegram (telebot) построил внутреннюю работу бота в приложении Telegram
- С помощью библиотеки pyzbar настроил механизм считывания QR-кодов из изображений посланных пользователями
- Создал базу данных с информацией о пользователях и книгах. Настроил взаимодействие между объектами внутри кода с помощью psycpg2

### Изображение проекции полиэдра | *Python, tkinter, ООП*

- Создал классы для удобной работы с объектами
- Использовал библиотеку tkinter для визуализации изображений

### Real-Time Chat Application | *Django, Channels, WebSocket, SQLite, Python, HTML/CSS*

- Реализовал систему обмена сообщениями в реальном времени через WebSocket с использованием Django Channels (асинхронные consumers)
- Спроектировал структуру БД на SQLite с моделями Room, Message и User, настроил CRUD-операции через Django ORM
- Разработал интерфейс чата с динамическим обновлением через JavaScript (без перезагрузки страницы)
- Реализовал формы аутентификации (регистрация/вход/выход) с использованием встроенной Django-логики

### Wildberries Product Position Parser | *Python, Requests, Telebot*

- Реализовал парсинг данных Wildberries с использованием библиотеки Requests (обработка JSON-ответов, извлечение ID товара)
- Разработал Telegram-бота с динамическим обновлением статуса поиска и обработкой пользовательских запросов через pyTelegramBotAPI
- Создал алгоритм генерации ключевых фраз из названий товаров (N-граммы 1-3 слов, фильтрация стоп-слов, частотный анализ через Counter)
- Реализовал многостраничный поиск позиций товара с ограничением до 10 страниц, обработкой таймаутов и ошибок API
- Оптимизировал URL-кодирование запросов с помощью urllib.parse для корректной работы с кириллическими запросами

### Car Sharing Database Design | *PostgreSQL, Database Design*

- Спроектировал концептуальную, логическую и физическую модели базы данных для сервиса каршеринга
- Реализовал физическую модель в PostgreSQL: создание таблиц (пользователи, автомобили, договоры, страховки, повреждения), установка связей и ограничений целостности
- Обеспечил соблюдение бизнес-правил: уникальность автомобилей по VIN/номеру, проверка дат договора, ограничение на несколько активных аренд у пользователя

### Sentiment Analysis Web Application | *Django, transformers, torch, Cuda*

- Реализовал простое веб-приложение для анализа тональности текстов на Django с моделью машинного обучения для предсказания рейтинга и тональности отзывов с IMDB

### Численное решение уравнения теплопроводности | *C++, CUDA-ready, CMake, ParaView*

- Разработал программу для решения уравнения теплопроводности в 1D с использованием явной конечно-разностной схемы
- Спроектировал архитектуру: классы Mesh (расчётная сетка), Solver (вычислитель), State (состояние системы), VTKWriter (визуализация)
- Реализовал поддержку VTK-формата для визуализации результатов в ParaView
- Подготовил код для возможного переноса вычислений на GPU (CUDA): использование сырых указателей, разделение логики, двойная буферизация
- Настроил автоматическую сборку с помощью CMake, обеспечив кроссплатформенность

## ТЕХНИЧЕСКИЕ НАВЫКИ

**Языки программирования:** Python, C/C++, Java, SQL (Postgres)

**Фреймворки:** Django, pyTelegramBotAPI, FastAPI, Flask

**Инструменты разработчика:** Git, Google Collab, VS Code, IntelliJ, CLion, pgAdmin

**Библиотеки:** Requests, pandas, NumPy, Matplotlib, tkinter, pyzbar, psycpg2, cv2, torch, transformers, sklearn