

# Тюпин Георгий

tyupin.georgy@gmail.com • Telegram: @georgetyupin • Москва, Россия

---

## О себе

Программированием занимаюсь с 12 лет, сейчас мне 20, так что можно сказать, что это мечта с детства. Бэкендом занимаюсь четыре года, Go пишу полтора. За это время успел поработать коммерчески, сделать несколько своих проектов и поконтрибьютить в open source.

Попробовал как фронтенд стек, так и бэкенд. Во фронтенд погружён достаточно базово. В бэкенде успел попробовать PHP, Python и Go. Решил выбрать бэкенд, так как нравится работать с бизнес-логикой и данными. А именно Go, потому что получаю удовольствие от разработки на нём. Нравится простой синтаксис, который позволяет реализовывать действительно производительные сервисы, нравится реализация многопоточности. Также было интересно изучать рантайм языка, разбираться в планировщике и GC.

Люблю разбираться в том, как устроены вещи внутри: как данные проходят через систему, где возникают узкие места, как правильно организовать код чтобы его было не стыдно читать через полгода. Умею писать чистую архитектуру, покрывать код тестами и работать с конкурентностью в Go.

Хочу попасть на стажировку Go-разработчика, где можно работать с реальными backend-задачами: сервисы, API, базы данных, архитектура. Это то место, где я смогу развиваться очень быстро.

---

## Опыт работы

### Backend-разработчик | Простая наука

Февраль 2024 — Октябрь 2024 | Москва

Разработал с нуля систему сегментации и векторизации изображений для производства DIY-наборов, обрабатывающую до 5000 изображений в день.

- Спроектировал микросервисную архитектуру: Python-сервис для ML-обработки (PyTorch, OpenCV, ComfyUI) и Go-сервис для управления файлами и координации между сервисами.
- Разработал REST API на Go для управления файлами с PostgreSQL для хранения метаданных.
- Написал Telegram-бота на Go для доставки готовых материалов внутренним пользователям.
- Контейнеризировал компоненты в Docker.
- Покрыл критические компоненты unit и интеграционными тестами.

**Стек:** Go, PostgreSQL, Docker, Git, Linux, Telegram Bot API, Python, PyTorch, OpenCV, ComfyUI

**Репозиторий:** [github.com/GeorgeTyupin/vectorizer](https://github.com/GeorgeTyupin/vectorizer)

---

## Open Source

### Вклад в Prometheus ([github.com/prometheus/prometheus](https://github.com/prometheus/prometheus))

Контрибьютил в open source инструмент мониторинга на Go.

- Исправлял баги из открытых issues: починил парсинг флагов командной строки, добавив корректные дефолтные значения.
  - Улучшал документацию: уточнял описания параметров конфигурации и поведение в граничных случаях.
  - Работа с большой кодовой базой дала понимание того, как организован Go-код в реальных высоконагруженных проектах.
-

## Проекты

### Judo Archive: Архив турниров по дзюдо

#### Репозитории:

- [github.com/GeorgeTyupin/judo\\_stats\\_site](https://github.com/GeorgeTyupin/judo_stats_site)
- [github.com/GeorgeTyupin/judo\\_parse](https://github.com/GeorgeTyupin/judo_parse)

Система сбора, обработки и хранения данных о соревнованиях по дзюдо с веб-интерфейсом для просмотра статистики.

- Написал парсер Excel-таблиц с нестандартной структурой: параллельная обработка файлов через горутины.
- Реализовал алгоритм транслитерации имён спортсменов с кириллицы на латиницу и четыре типа поиска дублей (точное совпадение, расстояние Левенштейна, подстрока, комбинированный).
- Сделал миграцию разобранных данных из Excel в PostgreSQL на удалённом сервере через SSH-туннель.
- Спроектировал схему БД для хранения профилей спортсменов, результатов турниров и весовых категорий.
- Реализовал REST API на Chi router со слоистой архитектурой: handlers / services / repository.
- Добавил server-side rendering на Templ с частичными обновлениями через HTMX, без отдельного фронтенд-фреймворка.

**Стек:** Go, Chi, PostgreSQL, Templ, HTMX, Docker, SSH, excelize, алгоритмы нечёткого поиска

---

### FileGuard: Платформа лицензирования и защиты цифровых продуктов

#### Репозиторий: [github.com/GeorgeTyupin/fileguard](https://github.com/GeorgeTyupin/fileguard)

Платформа лицензирования и защиты программных продуктов с проверкой доступа через десктопный клиент.

- Спроектировал систему из трёх компонентов: Telegram-бот для покупки лицензий, HTTP API сервер и десктопный клиент-верификатор.
- Реализовал REST API на Chi router: регистрация, верификация лицензий ( `POST /api/v1/verify` ), каталог продуктов.
- Разработал десктопный клиент (exe), который генерирует fingerprint устройства и проверяет лицензию перед запуском, возвращает exit code 0/1.
- Применил Clean Architecture: разделение handler -> service -> repository с Dependency Injection.
- Реализовал graceful shutdown.
- Спроектировал схему БД с таблицами users, products, purchases, devices и транзакционными операциями.

**Стек:** Go, Chi router, PostgreSQL (pgx), Telegram Bot API (telebot.v4), Docker, Clean Architecture

---

### MLCinema: Онлайн-кинотеатр с семантическим поиском

#### Репозиторий: [github.com/GeorgeTyupin/MLCinema](https://github.com/GeorgeTyupin/MLCinema)

Веб-приложение онлайн-кинотеатра с реализованным семантическим поиском фильмов.

- Реализовал backend на Go с Echo framework и GORM для работы с PostgreSQL.
- Спроектировал БД для фильмов, актёров и жанров.
- Реализовал Python/Flask микросервис с NLP-пайплайном: векторизация описаний через `distiluse-base-multilingual-cased-v1`, извлечение именованных сущностей через `natasha`, гибридный скоринг на основе косинусного расстояния и entity-matching.
- Создал REST API с эндпоинтами для поиска фильмов и фильтрации по категориям.

**Стек:** Go, Echo, GORM, PostgreSQL, Python, Flask, sentence-transformers, natasha, REST API

---

### Симулятор гравитации (курсовая работа)

**Репозиторий:** [github.com/GeorgeTyupin/stars-creation](https://github.com/GeorgeTyupin/stars-creation)

Курсовая работа, симулятор гравитационного взаимодействия небесных тел с визуализацией.

- Реализовал моделирование взаимодействия множественных тел на основе закона всемирного тяготения Ньютона.
- Создал графический интерфейс на Qt с цветовой градацией объектов по массе.
- Разработал предустановленные сценарии: солнечная система, столкновение галактик, массивные объекты.
- Реализовал слияние объектов при сближении и отображение статистики системы (энергия, импульс, угловой момент).

**Стек:** C++11, Qt 5/6, численные методы

---

## Программа статистического анализа данных (курсовая работа)

**Репозиторий:** [github.com/GeorgeTyupin/algorithms-and-data-structures](https://github.com/GeorgeTyupin/algorithms-and-data-structures)

Курсовая работа, приложение для анализа экспериментальных данных с графическим интерфейсом.

- Реализовал оценку параметров распределений (нормальное и Вейбулла) методами MLE и MLS.
- Разработал статистические критерии: Граббса, Фишера, Стьюдента, ANOVA, Шапиро-Уилка, Уилкоксона.
- Применил метод оптимизации Нелдера-Мида для поиска параметров распределений.
- Интегрировал Python (NumPy, Matplotlib, SciPy) для построения графиков и визуализации результатов.

**Стек:** C++17, Boost Math, Qt 6, Python (NumPy, Matplotlib, SciPy)

---

## Технические навыки

- **Языки программирования:** Go, Python, C++, SQL
  - **Базы данных:** PostgreSQL, Redis
  - **Технологии:** Docker, Git, Linux, REST API
  - **Фреймворки:** Echo, Chi, GORM, Telegram Bot API, PyTorch, OpenCV, Qt
- 

## Образование

### Московский авиационный институт (МАИ)

Прикладная информатика, 2 курс

2024 — настоящее время