

# МОРДВИНЦЕВ ДАНИЛА

+7(921) 898-8207 ◊ Санкт-Петербург, Россия

[idanilamor6@gmail.com](mailto:idanilamor6@gmail.com) ◊ [GitHub](#) ◊ [Telegram](#)

## ОБРАЗОВАНИЕ

Бакалавр по направлению «Software engineering», Университет ИТМО

Окончание в 2027 году

**Релевантные курсы:** ООП на C#, Управление данными в микросервисах на C#, C++, Java, Алгоритмы и структуры данных, Базы данных, Операционные системы, Платформы и среды выполнения языков программирования, Телекоммуникационные системы и технологии.

## НАВЫКИ

Языки программирования	C#, C++, Java
Технологии	.NET (ASP.NET, EF Core), gRPC, Kafka, Prometheus, Grafana, Git, CI/CD, Docker, SQL, Алгоритмы и структуры данных, ООП, Паттерны проектирования
Базы данных	PostgreSQL
Тестирование	XUnit, Moq/NSubstitute
Гибкие навыки	Техники публичных выступлений

## ЗНАЧИМЫЕ ПРОЕКТЫ

### Консольное приложение — эмуляция банковской системы

[GitHub](#)

- Создал консольное приложение на C# для выполнения базовых банковских операций (регистрация клиентов, открытие счетов, просмотр баланса, переводы средств) с интерактивным интерфейсом на базе библиотеки Spectre.Console.
- Спроектировал архитектуру в соответствии с принципами Onion Architecture: бизнес-логика сосредоточена в насыщенных предметных моделях, а взаимодействие между слоями организовано через встроенный DI-контейнер .NET.
- Для персистентного хранения данных задействована PostgreSQL; доступ к данным реализован через паттерн «репозиторий» с использованием асинхронных операций.

### Система управления платной клиникой — ClinicDB

[GitHub](#)

- Спроектировал и реализовал систему управления платной клиникой (ClinicDB) для обработки комплексных данных о пациентах, медицинском персонале, расписании приёмов, оказываемых услугах, финансовых операциях и результатах диагностики.
- Разработал детальную реляционную модель с 18 сущностями и соответствующей ER-диаграммой, отражающей все бизнес-процессы клиники: от регистрации пациентов до управления инвентарём и обучением врачей.
- Развернул высокодоступную PostgreSQL инфраструктуру с использованием Patroni для автоматической репликации, etcd для координации, HAProxy для распределения нагрузки между узлами, а также Flyway для версионирования и миграции схемы БД. Контейнеризация реализована через Docker и Docker Compose.
- Подготовил Python-скрипты с использованием Faker для наполнения системы релевантными тестовыми данными, соответствующими медицинской предметной области.
- Организовал полный цикл мониторинга и поддержки: настроил сбор метрик через Prometheus с аналитической панелью Grafana, внедрил автоматизированное резервное копирование, выполнил стресс-тестирование приложения с помощью k6.