

# Backend-разработчик – Рытиков Артём

---

Тел.: +7 (999) 440-38-95

Email: [rytikovartem@mail.ru](mailto:rytikovartem@mail.ru)

Москва

## Образование

НИЯУ МИФИ, Институт интеллектуальных кибернетических систем

Прикладная математика и информатика

Бакалавриат, 2024–2028 (2 курс)

## Ключевые навыки

Языки: Go, C++, Python

Базы данных: SQL

Инструменты: Git, gtest, wxWidgets, Docker

Концепции: ООП, алгоритмы и структуры данных, разработка микросервисов

## Проекты

Микросервис для назначения ревьюеров (Go)

Разработал HTTP-сервис, автоматизирующий процесс назначения ревьюеров на Pull Request'ы в команде, что сократило время распределения задач и повысило прозрачность процессов. Реализовал переназначение ревьюеров, фильтрацию PR по пользователям и управление командами.

Технологии: Go, HTTP API, Docker

GitHub: <https://github.com/ArtyomRytikov/pr-reviewer-service>

Электронная очередь в деканат (Python)

Разработана система электронной очереди для деканата, которая позволяет студентам записываться на прием к сотрудникам, проверять статус очереди и управлять своей записью. Система использует PostgreSQL для хранения данных и Flask для обработки запросов. Для обеспечения безопасности используется JWT (JSON Web Tokens). Веб-интерфейс написан с использованием HTML и CSS.

Технологии: Python, PostgreSQL, Flask, JWT, HTML, CSS

GitHub: [https://github.com/ArtyomRytikov/queue\\_system](https://github.com/ArtyomRytikov/queue_system)

Графы и алгоритмы на графах с графическим интерфейсом (C++)

В этом проекте реализованы различные структуры данных, такие как Set, Binary Tree, Dictionary, Priority Queue, Dynamic Array и Linked List. также добавлены структуры для построения ориентированных и неориентированных графов. Возможности работы с графиками включают:

1. Строительство случайного графа с произвольным числом вершин и рёбер.
2. Добавление рёбер и вершин в граф.
3. Алгоритм Дейкстры для поиска кратчайшего пути между заданными вершинами.
4. Поиск компонент связности в графе, определение их количества и максимального

компоненты связности.

5. Поиск радиуса и диаметра графа.
6. Топологическая сортировка, включая поиск источников и стоков.
7. Проверка частичного порядка, визуализация диаграммы Хассе и поиск минимальных и максимальных элементов.

Так же реализован графический интерфейс GUI на wxWidgets, где присутствует визуализация каждой задачи.

Технологии: C++, wxWidgets, gtest

GitHub: <https://github.com/ArtyomRytikov/lab3>

Структуры данных и задачи с графическим интерфейсом (C++)

В этом проекте реализованы различные структуры данных, такие как Set, Binary Tree, Dictionary, Priority Queue, Dynamic Array и Linked List. Выполнены следующие задачи:

1. Приоритизация элементов в очереди.
2. Поиск подстрок с возможностью анализа файлов и выбора диапазона длины строки для поиска.
3. Построение гистограммы по входной строке.

Так же реализован графический интерфейс GUI на wxWidgets, где присутствует визуализация каждой задачи

Технологии: C++, wxWidgets, gtest

GitHub: <https://github.com/ArtyomRytikov/lab2>

Система линейных структур данных (C++)

Создана модульная библиотека структур данных с поддержкой операций Map/Reduce/Zip, что позволяет унифицировать работу со структурами данных и улучшить тестируемость кода. Так же реализован графический интерфейс GUI на wxWidgets, где присутствует визуализация каждой задачи

Технологии: C++, gtest, wxWidgets

GitHub: <https://github.com/ArtyomRytikov/lab2-abstract-data-type>

Полиморфная система работы с матрицами (C++)

Реализована универсальная система работы с матрицами (целые, вещественные, комплексные числа) с поддержкой операций сложения, умножения, норм, определителя и преобразований строк/столбцов. Так же реализован графический интерфейс GUI на wxWidgets, где присутствует визуализация каждой задачи

Технологии: C++, gtest, wxWidgets

GitHub: <https://github.com/ArtyomRytikov/binary-tree-with-GUI>

Абстрактный тип данных на основе дерева (C++)

Разработано бинарное дерево с различными обходами и операциями (вставка, удаление, поиск, извлечение поддеревьев). Проведено тестирование производительности на наборах данных до  $10^8$  элементов. Так же реализован графический интерфейс GUI на wxWidgets, где присутствует вкладка тестирования и вывода графика зависимости времени операции от количества входных данных для каждой операции.

Технологии: C++, gtest, wxWidgets

GitHub: <https://github.com/ArtyomRytikov/binary-tree-with-GUI>

## О себе

Интересуюсь backend-разработкой, архитектурой микросервисов и высоконагруженных систем. Люблю разрабатывать структурированные и тестируемые решения. Постоянно изучаю новые технологии и улучшаю навыки написания эффективного кода.