

# Ivan Kosolapov

GitHub: [Zymik](#)

Telegram: [@xvanonex](#)

Email : [vanya.kosolapov.02@mail.ru](mailto:vanya.kosolapov.02@mail.ru)

Mobile : +7-911-322-94-60

Russian (Native), English (Intermediate)

## EDUCATION

- ITMO University** Saint-Petersburg, Russia  
*Bachelor in Applied Mathematics and Informatics*  
Sep. 2020 – Present

## WORK EXPERIENCE

- Yandex** Moscow, Russia  
*Java Developer Intern, Kinopoisk Showcase Team*  
July 2022 - Nov. 2022
  - Java
  - Spring Boot
  - Spring MVC
  - Flyway
  - SQL
  - Docker
  - Junit
  - Mockito
  - GraphQL
  - Implemented support of new device features configuration format
  - Added flyway migrations to one service
  - Created client for mocking service answer by value in JSON storage
  - Stress tested few services, added preheat to them, modified thread pool
  - Did many refactoring and code quality improvements: changed code for increasing integration tests coverage; added support of new version of domain objects, deleted all legacy code that used old domain objects
  - Researched anomaly request count to endpoint. Created convenient monitoring graphics
  - Did others minor tasks like adding new field to GraphQL data, integration tests rewrite etc.
  - Fixed many small bugs

## RECENT PROJECTS

- Post cat - posting application**
  - Scala
  - Functional Programming
  - Cats Effect
  - Cats
  - Http4s
  - Doobie
  - Tapir
  - ScalaTest
  - ScalaMock
  - The main idea of the project is to create an application for making posts on different platforms with telegram bot as user interface. Now implemented: core service for managing posts and groups; telegram bot to make posts to public channels; Public API for getting posts of groups. Also implemented commands DSL based on cats-parse to easily parse telegram messages to bot commands
- LL1 parser generator**
  - Rust
  - Parsers
  - Code generation
  - GraphViz
  - Nom
  - LL1 parser generator that implement part of popular [ANTLR](#) generator, but generate code in Rust that ANTLR doesn't support. Generator supports synthesized and inherited attributes. You can set tokens as regex for lexical analysis and characters to skip. Used [nom](#) as library of parser combinators to parse grammar description. Syntax tree can be easily visualized with [GraphViz](#)

## ACHIEVEMENTS

- ESI 19: international expo for young researchers - Participant** Abu-Dhabi, UAE  
Sep. 2019
  - Python
  - Arduino
  - Math modeling
  - Created prototype of controller based on Arduino for specific ([azipod](#)) kinds of ship. Created small simulator game that use that controller to control ship.

## RELEVANT UNIVERSITY COURSES

- Java Advanced**
  - Java
  - Concurrency
  - Reflection
  - NIO
  - IO
  - Collections
  - Streams
  - Network
  - Javadoc
  - Course is focused on advanced features of Java. On this course I completed several small projects including default interface implementation generator, web crawler that downloads web pages in parallel, simple non-blocking UDP server and client that uses Java NIO channels and etc. You can find my solutions on GitHub (link in header)
- Web Programming**
  - Java
  - Spring Boot
  - Spring MVC
  - Spring Data
  - Java Script
  - Vue.js
  - SQL
  - This course is introduction to web development that shows web technologies in progression: from Java servlets and HTML templates to Spring Boot and single page applications with Vue.js. You can find my solutions on GitHub (link in header)
- Functional Programming**
  - Functional Programming
  - Haskell
  - Course is focused on functional programming concepts and principals. It covers topics from algebraic data types and ends with more advanced constructions like Free Monad and Lens. You can find my solutions on GitHub (link in header)

## PROGRAMMING SKILLS

- Languages:** Java, Scala, Rust, Haskell, Python
- Frameworks & Tools:** Spring Boot, Spring MVC, Cats Effect, SQL, Bash, Flyway, Git, Docker
- Knowledge:** Type Theory, Functional Programming, Algorithms, Data Structures, Parallel Programming