Kosolapov Ivan

GitHub: Zymik
Telegram: @xvanonex
Email : vanya.kosolapov.02@mail.ru
Mobile : +7-911-322-94-60

Education

- **ITMO University** Saint-Petersburg, Russia
  Bachelor of 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 improvments: change code for increasing integration tests coverage; add support of new version of domain objects, delete all legacy code that used old domain objects
  - Researched anomaly request count to endpoint. Created convenient monitoring graphics
  - Did many small 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 project to create an application for making post on different platform with telegram bot as user interface. Now implemented: core service for managing posts and groups; telegram bot for post to public channels; public api for getting post of groups. Also implemented commands dsl based on cats-parse, that use to easily parse telegram messages to bot commands

- **LL1 parser generator**
  Rust | Parsers | Code generation | GraphViz | Nom

  Wrote LL1 parser generator that implement part of popular ANTLR generator, but generate code in Rust that ANTLR doesn't support. Generator support synthesized and inherited attributes. You can set tokens for lexical analysis and characters to skip as regex. 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
  Python | Arduino | Math modeling *Sep. 2019*

  Created prototype of controller based on Arduino for specific(azipod) kinds of ship. Created small simulator game that use that controller to control ship.

Related University Courses

- **Java Advanced**
  Java | Concurrency | Reflection | NIO | IO | Collections | Streams | Network | Javadoc

  Course focused on advanced features of Java and deep in common ones. Through this course I wrote many things like default interface implementation generator; web crawler, that download web pages parallel; simple non-blocking udp server and client, that use Java NIO channels and etc. You can look on my solutions in 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 show web technologies in progression: from Java servlets and html templates to Spring Boot and single page applications with Vue.js. You can my solutions in link in header

- **Functional Programming**
  Functional Programming | Haskell

  Course tells about functional programming conceptions and principals. Course starts from simple algebraic data types and end with more advanced constructions like Free Monad and Lens. You can look on my solutions in 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.