

Университет ИТМО, факультет программной инженерии и компьютерной техники  
Двухнедельная отчётная работа по «Информатике»: аннотация к статье

Выполнил(а) Крутько Н.А., № группы Р3113, оценка \_\_\_\_\_  
Фамилия И.О. студента не заполнять

**Название статьи/главы книги/видеолекции**

Доклад по Rust-Analyzer

**ФИО автора статьи (или e-mail)**

Алексей Кладов @matklad  
aleksey.kladov@gmail.com

**Дата публикации  
(не старше 2017 года)**

"15" сентября 2019 г.

**Размер статьи  
(от 400 слов)**

Презенташка, 43 стр.

**Прямая полная ссылка на источник и сокращённая ссылка (bit.ly, goo.gl, tr.im и т.п.)**

<https://github.com/rust-analyzer>

(Сокращённая, думаю, не требуется)

**Теги, ключевые слова или словосочетания**

Rust, IDE, compilers, IntelliJ Rust, JetBrains, Mozilla, Rust-Analyzer,

**Перечень фактов, упомянутых в статье**

1. В докладе было рассказано о уже существующей и довольно хорошей IDE для языка Rust: IntelliJ Rust, т.к. автор до этого принимал непосредственное участие в его разработке. IntelliJ Rust: отличная IDE, фронтенд компилятора Rust написан на Kotlin, GUI на Swing. Но написан не на Rust (хоть это и не сильно важно), компилятор — альтернативная реализация компилятора rustc.
2. Упомянуто про rustc: хороший компилятор (быстрая компиляция), но плохая IDE (довольно долгий отклик автокомплита).
3. Рассказано про основные идеи написания IDE: инкрементальность (переиспользование предыдущих результатов при незначительном изменении выхода) и ленивость (не производить вычисления, пока совсем не потребуется). Упомянуты основные результирующие: «- -»: простой компилятор, «+ -»: продвинутый компилятор, «- +»: простая IDE, «+ +»: rust-analyzer, rustc(?).
4. Описана специфика разрешения имён при ленивом (lazy) и при энергичном (eager) анализе кода, а также их плюсы и минусы. В IDE используется агрессивная ленивость: O(1) работы.
5. Описана специфика инкрементальности и почему не подходит для IDE: O(N) работы.
6. Описана специфика написания IDE под Rust: разрешение имён, макросы, из чего делаем вывод, что инкрементальность обязательна. Поэтому rust-analyzer ленив и для разрешения имён в модулях, инкрементален.
7. Рассказано о библиотеке Salsa для ленивых и икрементальных вычислений, которая позволяет добиться значительного прироста производительности и при этом позволяющая делать корректный анализ данных.
8. Описана специфика написания IDE на Rust. По словам автора он является идеальным языком для написания компиляторов, т.к. совмещает в себе преимущества C++ (стандарта для написания промышленных компиляторов) и OCaml (языка для компиляторов по определению), позволяет контролировать мутабельность и не потокобезопасные данные, модульность для больших проектов. Но так же это имеет и минусы: время компиляции и сложность поддержки IDE.

**Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)**

1. Rust является очень хорошим языком для написания компиляторов.
2. IntelliJ Rust является отличной IDE, но имеет существенные минусы.
3. Rust-Analyzer имеет все шансы стать отличной IDE.

**Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)**

1. Для написания IDE на Rust требуется довольно много человеко-часов.
2. Сложность поддержки IDE на Rust.
3. Сложности с решением специфичных вещей: разрешением имён, макросов.

**Ваши замечания, пожелания преподавателю или анекдот о программистах**

Найти новые статьи по системам счисления довольно сложно, т.к. это уже довольно старая и заезженная и тема (( $\int \square^\circ$ ) $\int \text{L}$ ). Поэтому доклад с последнего Rust Meetup в СПб.