

РАЗРАБОТКА ФРОНТЕНДА ПЛАГИНА ДЛЯ СТАТИЧЕСКОГО ВЫВОДА ТИПОВ В ЯЗЫКЕ RUBY

Автор — А.П. Кузнецов

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
телефон: +7 (931) 361-17-67, e-mail: alexvangogen@gmail.com

Научный руководитель — к.ф.-м.н. С.В. Григорьев
Санкт-Петербургский государственный университет

Ruby — язык программирования с неявной динамической типизацией. Как и для всех остальных языков с аналогичной типизацией, для Ruby характерны ошибки, связанные с неверными предположениями о типах, которые будут обнаружены непосредственно на этапе выполнения. Получение статической информации о типах помогло бы решить проблему, так как о типах внутри программы стало бы известно заранее, что позволило бы компиляторам лучше оптимизировать код, а интегрированным средам разработки (IDE) — строить новые виды статического анализа и давать больше подсказок пользователю языка при написании кода.

Существуют различные алгоритмы вывода типов для языков с динамической типизацией, однако они могут лишь уточнить типовую информацию — добиться полноты анализа типов в языке с динамической типизацией невозможно. Тем не менее, эта информация могла бы использоваться в IDE. Таким образом, **целью данной работы** является разработка плагина к IDE IntelliJIDEA / RubyMine, который дополняет исходный код типовой информацией и отображает эту информацию пользователю.

Для данного плагина разработана архитектура, представленная на рисунке 1.

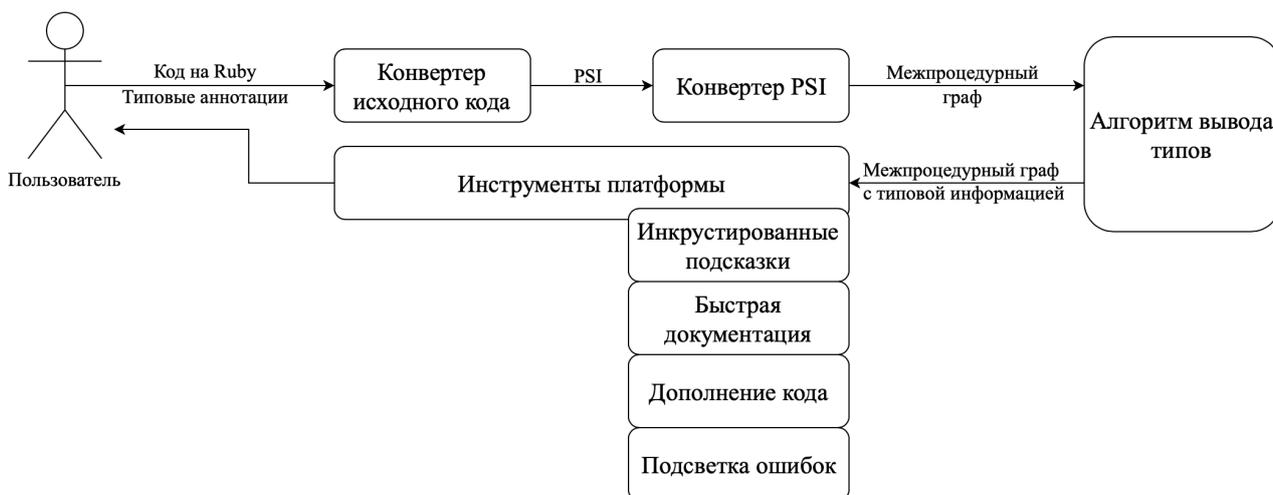


Рисунок 1. Архитектура плагина

Работа плагина выглядит следующим образом: пользователь пишет код на Ruby и может добавлять типовые аннотации. Далее, исходный код на лету переводится в PSI (Program Structure Interface) — специальное представление абстрактного синтаксического дерева. Для элементов PSI, соответствующих методам и выражениям, добавляется дополнительное свойство — тип этого элемента. Изначально тип известен у проаннотированных элементов, а также у элементов, тип которых тривиален, например, у целочисленных литералов.

На следующем шаге PSI переводится в межпроцедурный граф для подачи на вход алгоритму вывода типов. Вершины этого графа содержат информацию об элементах PSI исходного кода, например, позицию в тексте кода. Информация о типах элементов сохраняется.

В результате своей работы алгоритм вывода типов дополняет межпроцедурный граф той типовой информацией, которую ему удалось вывести. Наконец, плагин обратно сопоставляет вершины графа с элементами PSI, чтобы для них можно было получить поддержку со стороны IDE:

- отображение типов переменных/методов в редакторе;
- вывод информации о классе/переменной/методе во всплывающем окне при наведении;
- вывод на экран доступных пользователю атрибутов и методов класса при использовании выражения, которое имеет тип этого класса;
- подсветка ошибок при вызове метода, если в него переданы аргументы некорректных типов.

Таким образом, в ходе работы была разработана архитектура плагина, реализованы его компоненты, а также реализовано отображение типовой информации пользователю.