

УДК 004.432.2:004.4'232

СТАТИКО-ДИНАМИЧЕСКИЙ АНАЛИЗ S4 КЛАССОВ В ЯЗЫКЕ R

Тепляков В.В. (Университет ИТМО)

Научный руководитель – кандидат технических наук, доцент (квалификационная категория "ординарный доцент"), Корнеев Г.А.
(Университет ИТМО)

Настоящая работа посвящена статико-динамическому анализу S4 классов в языке R. Полученная информация используется в интегрированной среде разработки для упрощения работы пользователя с данными конструкциями языка.

Введение. В R существует несколько конструкций, в той или иной мере реализующих принципы ООП. Одна из таких конструкций именуется "S4 Class". В самом языке нет специального синтаксиса для объявления классов и создания объектов. Все операции определены через функции в пакетах стандартной библиотеки. Язык R динамический, но несмотря на то, что даже создание классов производится с помощью вызова функций, S4 классы являются строго типизированными языковыми конструкциями, нежели многие другие конструкции языка. Самым популярным решением для написания кода на R является R Studio. R Studio предполагает, что пользователь часто взаимодействует с консолью, и поэтому большое количество информации берется из нее. В данной IDE есть простейшая поддержка S4 классов, такие как автодополнение полей объектов (только на основе информации из консоли), переход к определению метода (только в рамках одного файла и если это единственный метод с таким именем в файле) и др. Однако необходимость держать много информации загруженной в консоль вносит неудобства при работе с кодом. К тому же, при перезапуске консоли вся информация будет потеряна. В то же время, весьма строгий способ объявления S4 сущностей позволяет производить статический анализ существующих классов и сильно улучшать автодополнение, переход к определению, переименование и другие подсказки, предоставляемые IDE.

Основная часть. Важно понимать, кто является потенциальным пользователем языка. В первую очередь R — язык для статистической обработки данных и работы с графикой. Из-за этого, большая часть потребностей пользователей связана с классами, объявленными в сторонних пакетах, а не написанных самим пользователем. Все классы, объявленные в пакете, загружаются в структуры внутри интерпретатора при загрузке самого пакета. Доступ к этим структурам можно получить с помощью вызова служебных функций. Этот факт был использован для анализа классов сторонних библиотек: при первом запуске IDE все сторонние пакеты индексируются путем запуска интерпретатора, загрузки пакета и последующего сбора информации о всех классах этого пакета. Процесс индексации был подвергнут различным оптимизациям и требует небольшого количества времени. Полученной в результате индексации информации достаточно для реализации подавляющего функционала, который пользователь ожидает от интегрированной среды разработки. Индексация покрывает потребности большого количества пользователей языка, но никак не помогает в работе людям, которые пишут свои собственные классы. Из-за того, что S4 классы объявляются через вызовы функций, невозможно гарантированно определить даже их структуру статическим анализом кода. Поэтому, для анализа было принято решение использовать эвристики. В результате изучения открытого исходного кода было обнаружено, что очень малый процент объявлений классов содержит сложные конструкции языка или вызовы сторонних функций. Почти все объявления либо напрямую содержат перечисления полей, родительских классов, валидаторов и т.д., либо используют переменные, в которые необходимая информация записывается явно. Таким образом, используются следующие эвристики: 1) анализ аргументов функций объявления классов согласно распространенным

паттернам и 2) рассмотрение только явных аргументов и аргументов-переменных на глубину не более 1.

При использовании только описанного выше статического подхода некоторые подсказки становятся неактуальными, так как окружение консоли может сильно отличаться от полученной информации. При использовании только динамической информации из консоли невозможно предоставить сколько-нибудь качественные подсказки из-за ограниченности получаемой информации. Хороший результат дает сочетание обоих подходов: помимо информации, полученной статически, для улучшения подсказок, по возможности, необходимо использовать консоль (например, для сортировки результатов автодополнения).

Выводы. В рамках продукта RPlugin на основе IntelliJ Platform были реализованы описанные в работе эвристики и индексация для получения необходимой информации о существующих S4 классах. На основе полученных данных были реализованы автодополнение, переход к определению и использованию, переименование, подсветка предупреждений и ошибок для S4 классов. Для улучшения предоставляемых подсказок также активно используется информация из консоли. Если статическая и динамическая информация конфликтуют, приоритет отдается определенному результату в зависимости от контекста подсказки.

Тепляков В.В. (автор)

Подпись

Корнеев Г.А. (научный руководитель)

Подпись