



Язык Рефал Плюс

Язык Рефал Плюс



Язык программирования Рефал разработан В.Ф. Турчиным в шестидесятые годы прошлого столетия как функциональный язык, ориентированный на задачи, связанные с преобразованием символьной информации (symbol manipulation).

Рефал Плюс — это диалект языка Рефал, разработанный Р.Ф. Гуриным и С.А. Романенко в 1991-ом году. В 2006-ом году синтаксис языка был адаптирован для более гладкого взаимодействия с современными программными средами. Инструменты для программирования на нем включают полноценную интегрированную среду разработки на базе платформы Eclipse. Современная реализация Рефала Плюс (см. рис. 1) базируется на компиляции программного кода на Ре-фале в код на широко доступных языках программирования, таких как C++ и Java, байткод для виртуальной машины Java, а также язык T++ — расширение C++, используемое в качестве основного языка в системе динамического распараллеливания программ OpenTS, предназначенной для исполнения на суперкомпьютерах.

Высокая модульность реализации позволяет также интегрировать в систему другие диалекты языка Рефал, такие как Рефал-5 и Рефал-6, обеспечивая прозрачное взаимодействие кода на них.

Рефал имеет большое преимущество перед многими другими языками программирования при решении задач символьной обработки. Однако, набор библиотек для всех имеющихся диалектов Рефала достаточно беден по сравнению с более популярными языками. Исходя из этих соображений, продуман и реализован механизм взаимодействия программ на языках Рефал Плюс и Java. В рамках единой однородной среды Eclipse стало возможно реализовывать проекты, совместно используя эти языки, а также и другие языки, работающие на платформе Java.

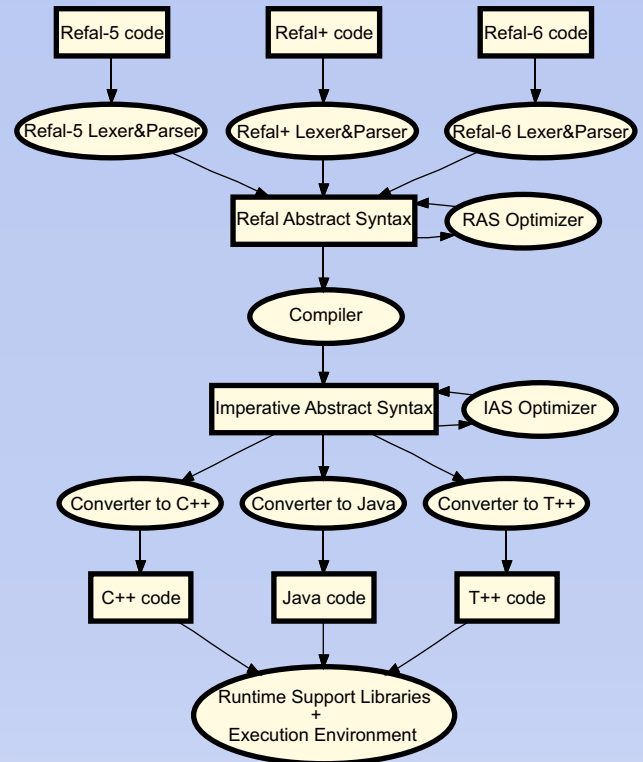


Рис. 1 Схема современной реализации языка Рефал

Средства для работы с Рефалом в среде Eclipse

Современные интегрированные среды разработки отличаются от текстовых редакторов прежде всего тем, что оказывают поддержку программисту на уровне семантики используемых языков программирования. Развитые текстовые редакторы могут подкрасить исходный код на основе синтаксического анализа программы, однако на основе семантического анализа можно предоставить программисту гораздо более полную информацию.

Помимо интеллектуальной раскраски исходного текста программы, семантический анализ дает возможность производить:

- навигацию по программе;
- осмысленное форматирование кода;
- подсказывать свойства программных объектов;
- изображать в реальном времени ошибки прямо в коде программы;
- производить автоматизированную рефакторизацию (изменение кода без изменения смысла программы)
- и многое другое.

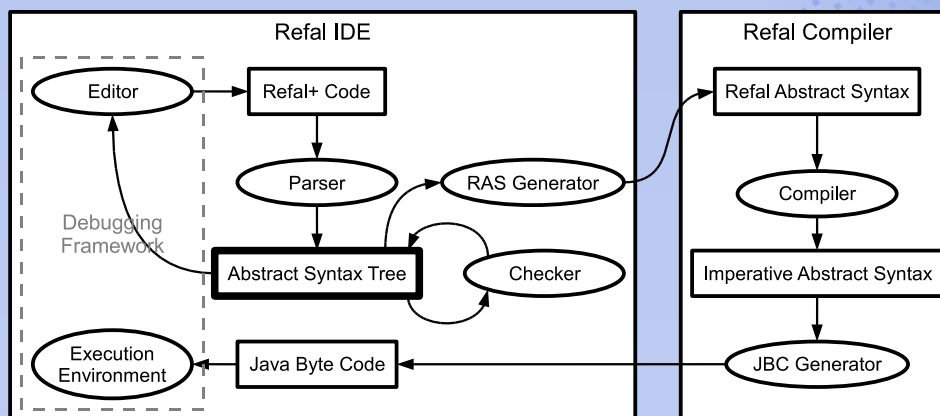


Рис. 2. Схема работы IDE для языка Рефал

С целью реализации описанных возможностей для языка Рефал Плюс, создана и развивается интегрированная среда разработки на базе платформы Eclipse. Использование такого мощного фундамента позволило сократить до минимума затраты на повторное программирование стандартных средств (интерфейса, окон, и т. п.) и сосредоточить усилия на специфических для языка Рефал направлениях. Кроме того, использование унифицированного интерфейса среды разработки облегчает жизнь программисту.

Основным инструментом взаимодействия между средой и программистом является редактор программ (Editor, см. рис. 2). Непосредственно во время написания программистом кода, при помощи парсера (Parser) строится объектное представление абстрактного синтаксического дерева программы (AST). Особенностью парсера является устойчивость к синтаксическим ошибкам в программе. Часть выражений может быть не дописана, но, несмотря на это, парсер построит AST, что позволит редактору отобразить семантическую информацию для корректных частей программы.

Над AST производятся различные проверки на корректность с точки зрения входного языка (Checker). В процессе выполнения проверок в AST добавляется новая семантическая информация, которая затем используется редактором. Если все проверки прошли успешно, то программа корректна.

Для того чтобы программу можно было исполнить, AST компилируется в байт-код виртуальной машины Java (JBC). Для этого используется описанный выше универсальный компилятор Рефала в императивные языки. Тривиальным преобразованием AST отображается в абстрактный синтаксис Рефала (RAS), и далее используется компиляция RAS → IAS и конвертер в JBC в качестве бэкенда.

Интересно, что весь блок «Refal Compiler» на рисунке 2 реализован на языке Рефал Плюс. При этом он вызывается из плагина для Eclipse, реализованного на языке Java. Для генерации JBC код на Рефале Плюс вызывает стороннюю библиотеку ASM, реализованную на языке Java, и затем отдает обратно в IDE готовый байт-код для записи в class-файл. Из этой схемы видно, что тесная интеграция кода на языках Рефал и Java может быть

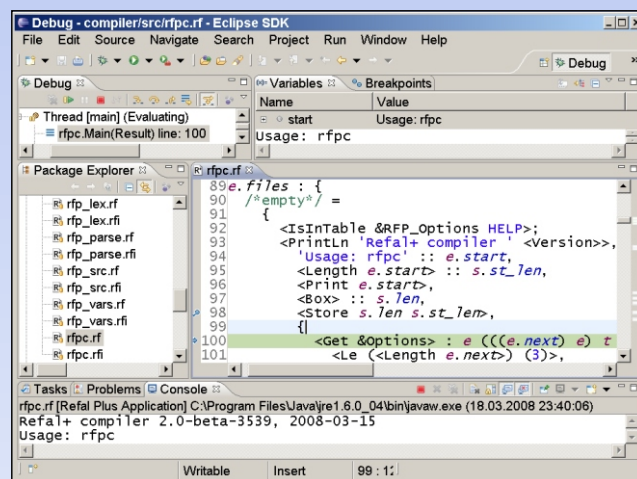


Рис. 3. Отладка программы на Рефале Плюс

Важным свойством среды разработки является наличие интерактивного отладчика, позволяющего отследить работу программы на уровне исходного кода. Во время отладки программист может следить за ходом выполнения программы непосредственно в окне редактора (см. рис. 3). В других окнах ему доступна информация о значениях переменных и стеке вызовов, ввод и вывод данных, список точек останова (breakpoints). Созданные средства позволяют удобно и эффективно разрабатывать программы на языке Рефал Плюс, а также легко интегрировать код на Рефале с кодом на других популярных языках в рамках единого программного продукта.