

Интеграция языка для работы с реляционными базами данных в базовый язык JetBrains MPS

Никитин Павел Антонович

Кафедра системного программирования, математико-механический
факультет СПбГУ, 444 группа

Научный руководитель: Вадим Гуров (кандидат наук,
программист ООО "ИнтеллиДжей Лабс")

На данный момент существует несколько классов решений для работы с реляционными базами данных из языка общего назначения (Java).

- ▶ Препроцессорные Embedded SQL, SQLJ в Java
- ▶ Слабо структурированный JDBC
- ▶ Внутренние DSL (как jequel)

Популярные решения (Embedded SQL) появились довольно давно и требуют от программиста повышенной внимательности, препроцессор не даёт продуктивности современных IDE.

Создание интегрированного в Java языка для работы с реляционными базами данных со структурированной подсказкой вводимого кода, подсветкой ошибок, анализом потока управления, статической типизацией рефакторингом и прочими полезными функциями.

Для создания языка был выбран языковой инструментарий JetBrains MPS (Meta Programming System) – средство для создания языков и интегрированная среда для разработки на них. Созданы:

- ▶ Иерархия концептов (возможных вершин AST – Abstract Syntax Tree) для SQL в MPS так, чтобы она была интегрирована в соответствующую иерархию в Java
- ▶ Текстово-подобные редакторы для них
- ▶ Дополнительная иерархию концептов (таких, как кортежи) для приведения данных SQL к baseLanguage и наоборот
- ▶ Комплементарная дополнительной иерархии библиотека
- ▶ Генератор новых конструкций в Java, JDBC и вызовы этой библиотеки
- ▶ Статическая система типов для проверки их на лету в IDE

В результате был получен расширяемый язык для работы с реляционными базами данных с полноценной IDE-инфраструктурой на базе JetBrains MPS, с интеграцией в Java, с широкой возможностью расширения, генерируемый в распространённый стандарт JDBC. Использование этого языка потенциально эффективнее существующих для решений встроенного SQL в Java.

Дальнейшие возможности улучшения:

- ▶ Добавление типизации для встроенных в SQL функций
- ▶ Оптимизация формирования запросов во время исполнения
- ▶ Проверка типов для вложенных SQL-запросов