

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра системного программирования

Поддержка консистентности модели семантической
подсветки в редакторе IDE при конкурентных
модификациях

Курсовая студента 1-го курса магистратуры, 546 группы
Кирсанова Александра

Научный руководитель: профессор Терехов А.Н.
Рецензент: руководитель отдела разработки
JetBrains, Сергей Шкредов

Введение

Семантическая подсветка является неотъемлемой частью функциональности, предлагаемой современными текстовыми редакторами и IDE. Подсветка различных идентификаторов позволяет разработчику быстрее ориентироваться в коде программы. Markup-model также выступает в роли одного из основных способов взаимодействия с пользователем и помогает сообщить программисту о ошибках и возможных недочетах и плохих практиках (code smells).

Маркап-модель должна корректно отражать состояние документа иначе в семантической подсветке не будет никакой пользы.

Постановка задачи

Структура маркап-модели

Если формализовать задачу, которую приходится решать IDE для предоставления таких возможностей, можно сказать, что маркап-модель представляет собой дополнительную древовидную структуру над текстом в документе. Узлами в дереве выступают отрезки, к которым в последующем будут применены различные визуальные эффекты. Отрезки для подсветки (или highlighters) добавляются в модель в ходе построения абстрактного дерева разбора и при выполнении различных анализов. Большая часть таких отрезков постоянно изменяет свои границы (например, при нажатии пробелов или знаков переноса строки в самом начале файла, изменяются границы абсолютно всех highlighter'ов, несмотря на то, что структура AST не изменилась). Удаление большего куска текста из редактора тоже приводит к изменению маркап-модели, редактор не может подсвечивать слова, если сами слова удалены.

Сохранение консистентности такой модели при документных изменениях решается внутри подсистем самого редактора (TextControls) и в контексте данной работы не представляет большого интереса. Гораздо более важным является добавление и удаление хайлайтеров в результате изменения структуры программы (PSI, Program structure interface).

Пример

Представим пример, мы работаем с проектом на языке C#, в одном файле используется множество ещё не объявленных классов, все эти использования подкрашены как ошибки, потом мы создаем соседний файл с тем же пространством имён (namespace) и объявляем в нём все недостающие классы и методы. Первый файл остался неизменным, но изменился PSI и подсвеченные ошибки больше не являются актуальной информацией о файле. Такое обновление не может быть сделано редактором с помощью анализа одних документных изменений. Изменения PSI происходит почти на любое изменение документов в проекте (кроме исключительных случаев со знаками табуляции и переноса строк в языках не чувствительных к отступам), поэтому любое изменение документа приводит к добавлению и удалению хайлайтеров.

Проблема

Если обновление документа и макрап-модели осуществляется синхронно в рамках одного процесса, то механизмов редактора и передачи координат отрезков, которые нужно добавить или удалить, достаточно для поддержки консистентности. Однако если эти условия не выполняются, и редактор находится в другом процессе от анализатора, а изменения документов и запросы на добавление/удаление хайлайтеров происходят асинхронно, макрап-модель очень быстро оказывается в некорректном состоянии.

Задача

Эта работа посвящена созданию механизма для передачи и удаления хайлайтеров в многопроцессорном приложении в проекте ReSharper, который бы обеспечивал консистентность макрап-модели при любых последовательностях изменений. Дополнительным требованием к данной работе является минимизация трафика, необходимого для передачи и поддержки консистентности модели.