

Визуализация осуществлённых рефакторингов в IDE на основе IntelliJ Platform

Соловьев Александр

группа 16.Б10-мм

руководитель: к.т.н., доц., Т.А. Брыксин

рецензент: Зам. начальника отдела разработки

ПО СОРМ ООО “НТЦ Протей”, Зимин А.С.

Санкт-Петербургский государственный университет

Введение

- Для улучшения исходного кода проводятся рефакторинги
- Знание о рефакторингов может быть полезно как для самих разработчиков, так и для каких-либо инструментов
- Есть разные подходы к поиску рефакторингов
- Нет ни одного из них интегрированного с IntelliJ Platform

Постановка задачи

Целью данной работы является разработка плагина для IDE на основе IntelliJ Platform, позволяющего находить рефакторинги в истории проекта

- Выбрать наиболее подходящий подход к поиску рефакторингов
- Спроектировать модульную архитектуру решения для поддержки работы с разными языками
- Реализовать интегрированную с IntelliJ Platform библиотеку, способную работать с разными языками
- Провести апробацию полученного решения

IntelliJ Platform

Основа для IDE от компании JetBrains и других разработчиков, предоставляющая одинаковый API для работы с ними

- Поддерживает плагины
- Использует PSI для представления синтаксических деревьев различных языков программирования
- Для работы с git использует git4idea

Сравнение методов

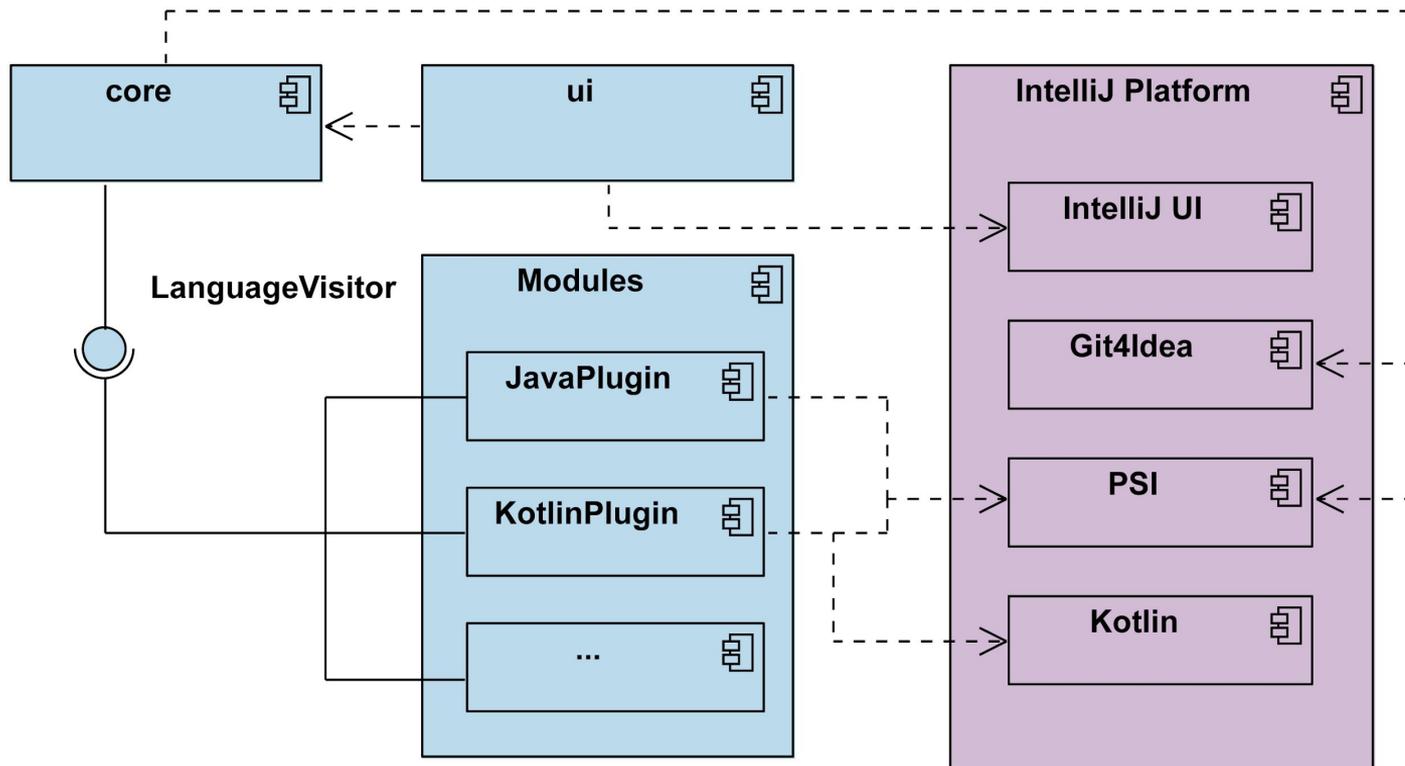
| Метод | Точность ¹ | Полнота ¹ | Нужен целый проект |
|--------------------|-----------------------|----------------------|--------------------|
| RefDiff | 1.000 | 0.877 | Нет |
| RMiner | 0.956 | 0.728 | Нет |
| RefactoringCrawler | 0.419 | 0.356 | Да |
| Ref-Finder | 0.264 | 0.624 | Да |

¹ – Silva D., Valente M. T. RefDiff: Detecting Refactorings in Version Histories // 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR).—2017.—P. 269–279.

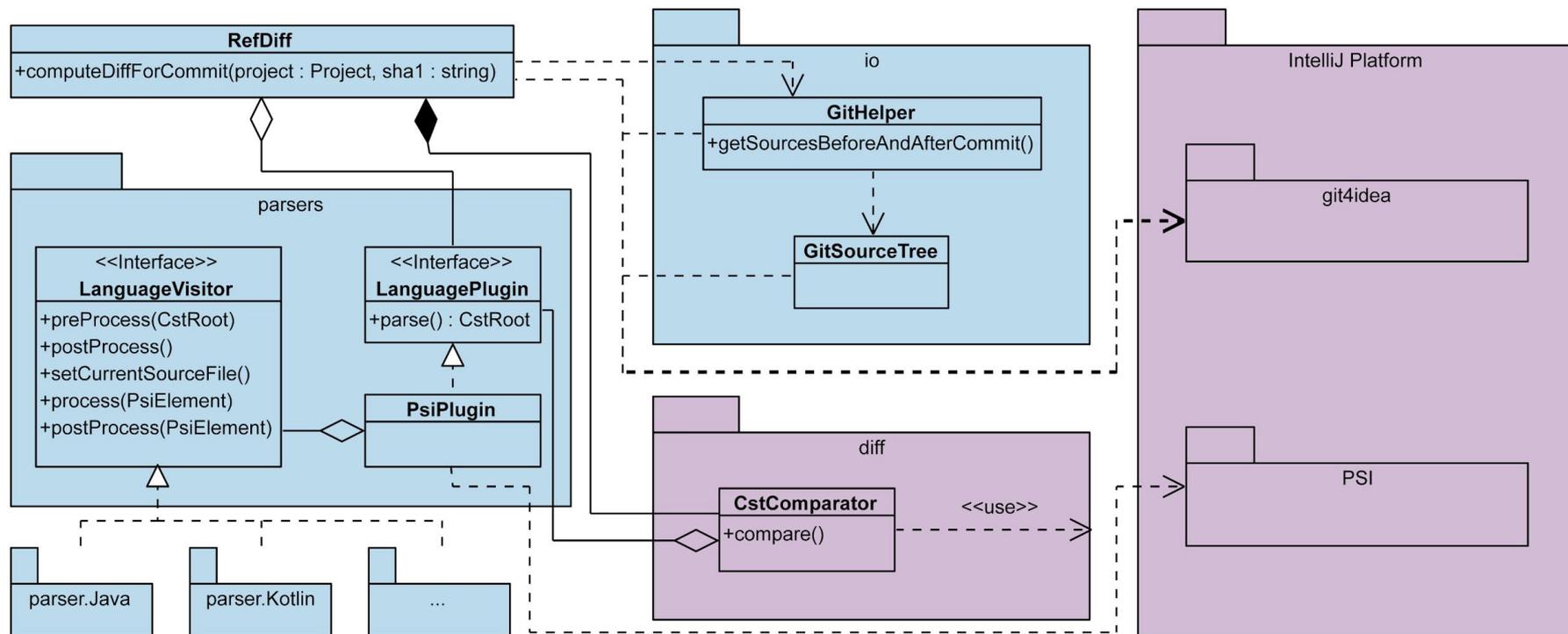
Сравнение времени работы в миллисекундах

| Репозиторий | Среднее | | Максимальное | |
|----------------------|-------------|--------|--------------|---------|
| | RefDiff 2.0 | RMiner | RefDiff 2.0 | RMiner |
| helidon | 174 | 418 | 21 983 | 112 807 |
| hutool | 81 | 34 | 1 833 | 782 |
| igniter | 15.6 | 14.8 | 75 | 343 |
| nacos | 65 | 234 | 2 076 | 133 827 |
| resilience4j | 79 | 146 | 11 756 | 30 207 |
| RxJava | 262 | 490 | 26 139 | 448 664 |
| spring-boot | 136 | 88 | 150 289 | 92 117 |
| spring-cloud-alibaba | 102 | 35 | 14 191 | 2 972 |
| vhr | 26.5 | 14.2 | 1 079 | 591 |

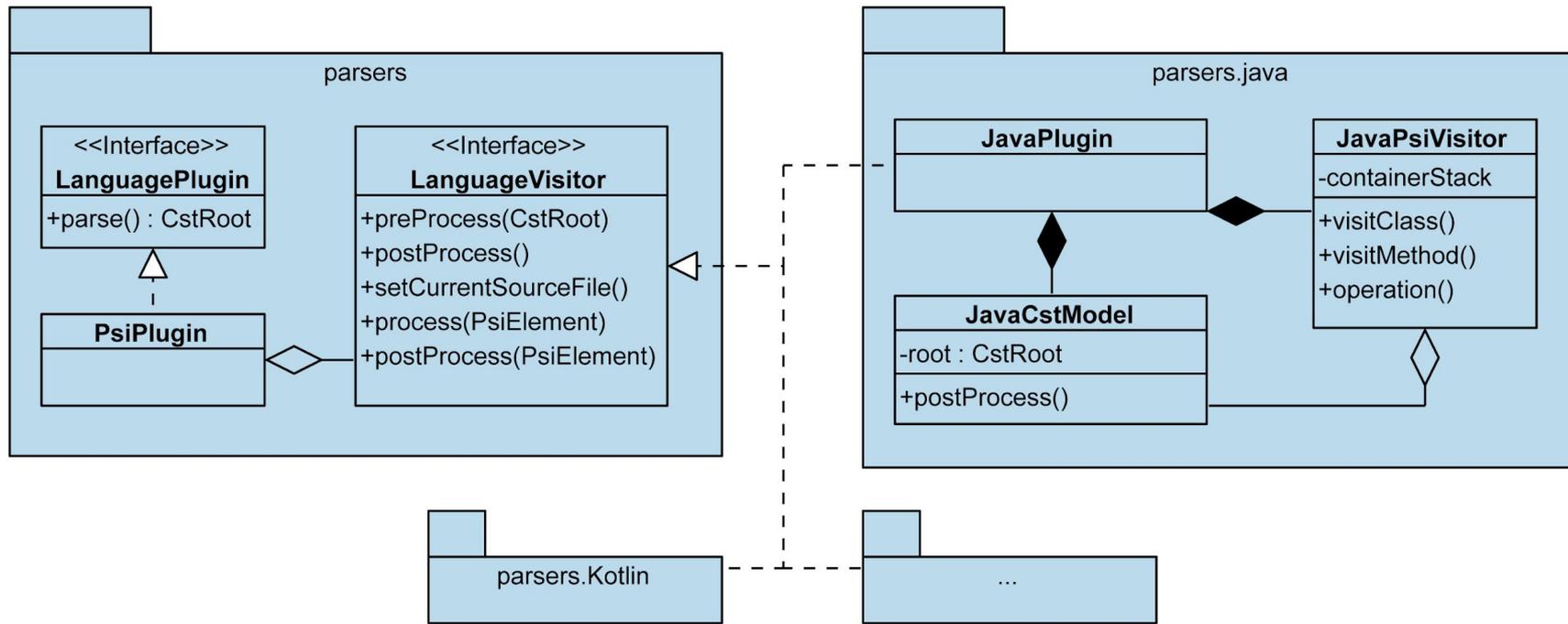
Диаграмма основных компонентов



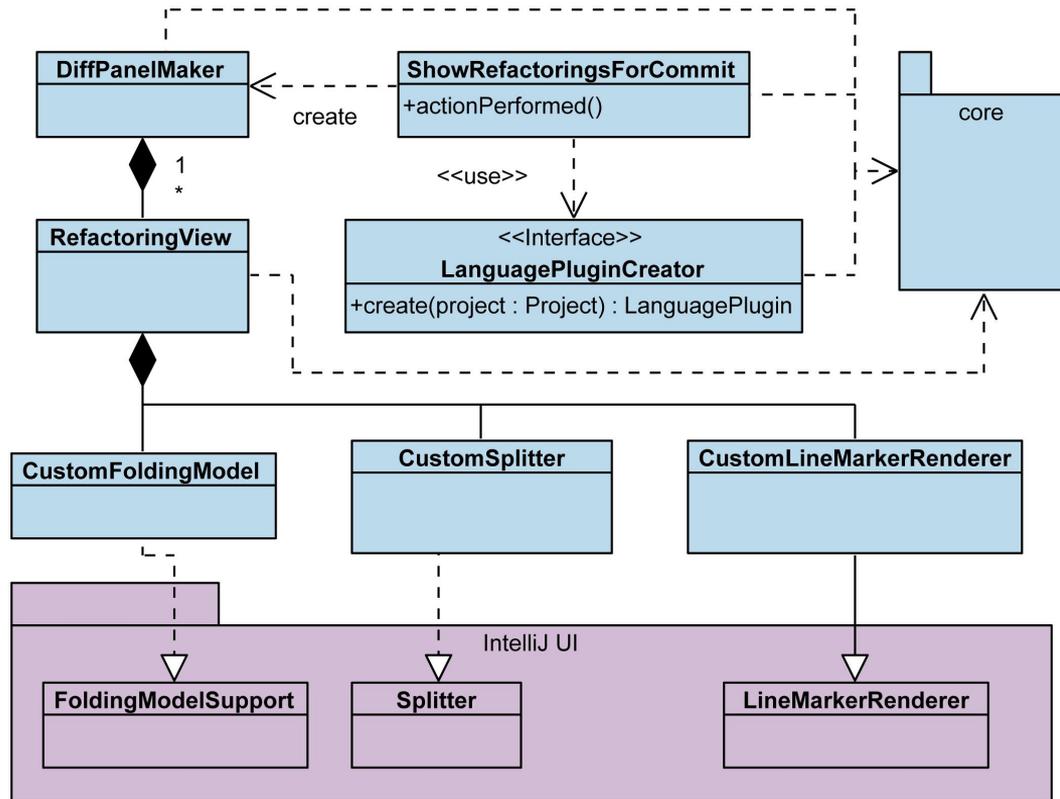
Архитектура основной компоненты “core”



Архитектура расширения для проектов на Java



Архитектура модуля визуализации



Визуализация извлечения метода в проекте на Kotlin

Extracted ClassA produceResult() from ClassA f()

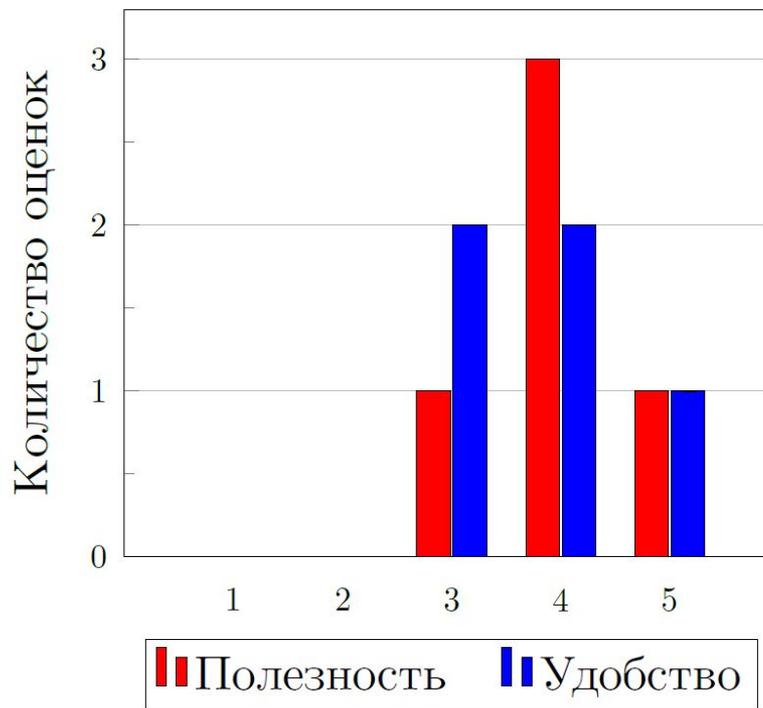
```
C:/Tst/KotlinTest/src/main/kotlin/ClassA.kt
class ClassA {
    fun f(x : Int) : Int {
        var result = x * x
        result += 24
        result -= 53
        helper(result)
        result -= 35
        return helper(result)
    }
}

C:/Tst/KotlinTest/src/main/kotlin/ClassA.kt
class ClassA {
    fun f(x : Int) : Int {
        var result = x * x
        return produceResult(result)
    }

    private fun produceResult(result: Int): Int {
        var result1 = result
        result1 += 24
        result1 -= 53
        helper(result1)
        result1 -= 35
        return helper(result1)
    }
}
```

Апробация

- Java
- 5 разработчиков
- Полезность — насколько качественно находит рефакторинги
- Удобство — насколько хорошо представлена информация



Результаты

- Выбран наиболее подходящий инструмент для поиска рефакторингов в истории проектов
- Спроектирована модульная архитектура решения, поддерживающая работу с разными языками
- Спроектированная архитектура реализована в качестве библиотеке, интегрированной с IntelliJ Platform
- Разработаны расширения для проектов на языках Java и Kotlin на основе реализованной архитектуры
- Проведена апробация решения