Функциональный рефакторинг клонов

Плагин для IDEA

Симиютин Борис
группа 504
руководитель М. Ахин

СПбАУ

26 декабря 2016 г.

Введение

- Бывает, функции в чем-то похожи, но есть принципиальные отличия, например, в предикатах.
- Захотелось попробовать доставать из них различия, и рефакторить получившиеся клоны.
- Для реализации пользовались встроенными механизмами рефакторинга IDEA

Алгоритм

- Для каждого файла:
  - Находим все места, где можно применить один из встроенных рефакторингов перехода от циклов к Stream API. Применяем их. Это сужает рассматриваемую область.
  - Внутри методов, которые затронул предыдущий шаг, ищем выражения, которые можно извлечь как фукциональные параметры. Извлекаем их.
  - По всему проекту ищем клоны в получившемся коде, удаляем полностью совпадающие методы, и заменяем вхождения в неполностью совпадающих.

Пример работы. Исходный код.

```
1  List<Invoice> getInvoicesFromSun(List<Invoice> invoices)
2  {
3      List<Invoice> result = new ArrayList<>();
4      for(Invoice invoice: invoices)
5          if(invoice.getCustomer() == Customer.SUN)
6              result.add(invoice);
7
8      return result;
9  }
10
11 List<Invoice> getInvoicesFromMS(List<Invoice> invoices)
12 {
13     List<Invoice> result = new ArrayList<>();
14     for(Invoice invoice: invoices)
15         if(invoice.getCustomer() == Customer.MICROSOFT)
16             result.add(invoice);
17
18     return result;
19 }
```

Пример работы. Применяем функциональные рефакторинги.

```
1  List<Invoice> getInvoicesFromSun(List<Invoice> invoices)
2  {
3      List<Invoice> result = invoices
4          .stream()
5          .filter(invoice ->
6              invoice.getCustomer() == Customer.SUN)
7          .collect(Collectors.toList());
8
9      return result;
10 }
11 List<Invoice> getInvoicesFromMS(List<Invoice> invoices)
12 {
13     List<Invoice> result = invoices
14         .stream()
15         .filter(invoice ->
16             invoice.getCustomer() == Customer.MICROSOFT)
17         .collect(Collectors.toList());
18
19     return result;
20 }
```

Пример работы. Рефакторим клоны.

```
1 List<Invoice> getInvoicesByPredicate(List<Invoice>
      invoices, Predicate<Invoice> invoicePredicate)
2 {
3     List<Invoice> result = invoices
4         .stream()
5         .filter(invoicePredicate)
6         .collect(Collectors.toList());
7
8     return result;
9 }
```

Моменты

Моменты

- Поведение в коде, бросающем исключения.
- Методы, реализующие интерфейсы.
- Ссылки на локальные переменные, вызовы локальных методов.
- Коллизии имен в вызывающем коде при подъеме параметров.
- Аккуратность при удалении полных клонов.

Допущения

Допущения

- Если в коде мало мест, где можно применить migration to streams, то мало чего найдем.

Допущения

- Если в коде мало мест, где можно применить migration to streams, то мало чего найдем.
- Во многих более-менее сложных ситуациях не проводим рефакторинг.

Допущения

- Если в коде мало мест, где можно применить migration to streams, то мало чего найдем.
- Во многих более-менее сложных ситуациях не проводим рефакторинг.
  - Если код бросает исключения - не проводим рефакторинг.
  - Если код находится в методе, реализующем интерфейс - не проводим рефакторинг.
  - Если в полученных лямбдах есть ссылки на локальные переменные, или вызовы методов текущего класса - не проводим рефакторинг.

Решения

- Коллизии имен в вызывающем коде при подъеме параметров - добавляем guid-соль.
- Нужна аккуратность при удалении полных клонов - удаляем аккуратно.

Проверка боем

- тестовый проект - Apache Ant

Проверка боем

- тестовый проект - Apache Ant
  - изменено 165 файлов

Проверка боем

- тестовый проект - Apache Ant
  - изменено 165 файлов
  - собирается (от одного рефакторинга и одного файла пришлось отказаться)

Проверка боем

- тестовый проект - Apache Ant
  - изменено 165 файлов
  - собирается (от одного рефакторинга и одного файла пришлось отказаться)
  - проходит встроенные тесты

Проверка боем

- тестовый проект - Apache Ant
  - изменено 165 файлов
  - собирается (от одного рефакторинга и одного файла пришлось отказаться)
  - проходит встроенные тесты
  - получилось забутстрапить

Проверка боем

- Но:

Проверка боем

- Но:
- Найденных клонов - 0

https://github.com/simiyutin/java_functional_clones