Улучшение автодополнения для языка Groovy в IDE IntelliJ IDEA

Курсовая работа студента 445 группы
Абишева Тимура Маратовича

Научный руководитель:
М. А. Мухин

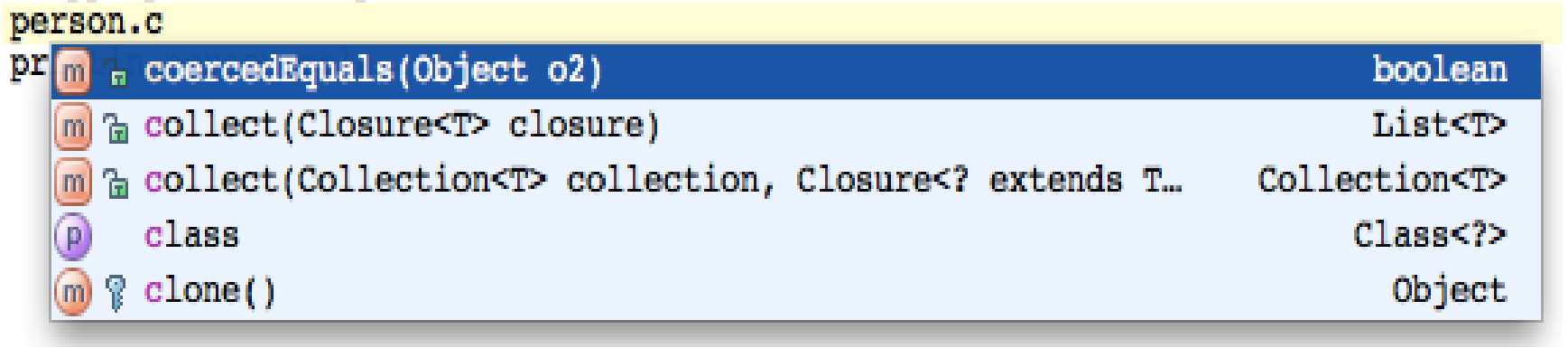
Санкт-Петербург, СПбГУ, 2011

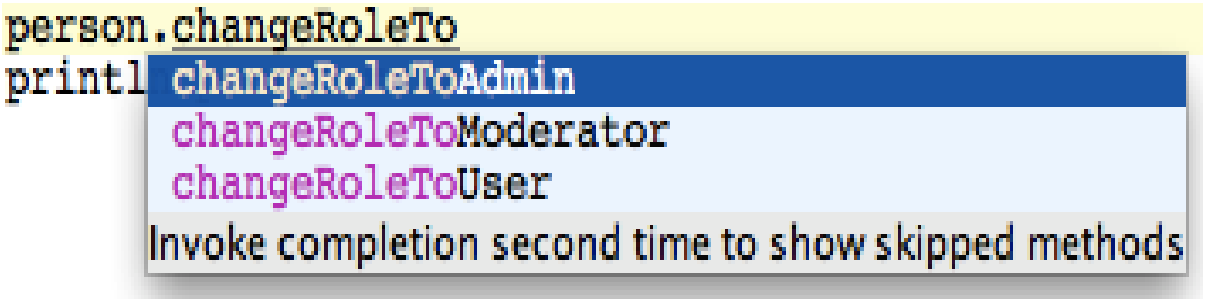
Введение

- Язык Groovy
  - Объектно-ориентированный
  - Динамическая типизация
  - Работает на JVM, есть черты от Ruby, Smalltalk, Python
- Grails
  - Веб-фреймворк, вдохновленный Ruby On Rails
  - Поддерживается такими гигантами, как SpringSource и SAP

Проблема

- Для динамических, а также активно использующих метапрограммирование, языков программирования автодополнение работает плохо





Задача

- Улучшить автодополнение в языке Groovy для часто встречающихся ситуаций
  - Использование метапрограммирования в контексте класса

```
class Person {
    String role = "admin"

    static {
        for (_newRole in ["admin", "user", "moderator"]) {
            def newRole = _newRole
            Person.metaClass."changeRoleTo${newRole.capitalize()}" =
                {-> delegate.role = newRole }
        }
    }
}
```

Причины плохого автодополнения

- Невозможно вывести тип переменной

```
def myTestMethod(someVariable) {
    someVariable.someMethod()
}
```

Причины плохого автодополнения

- Наличие метода method_missing
  - Используется, к примеру, для таких методов, как Person.findAllByName(name) в GORM*

Решение

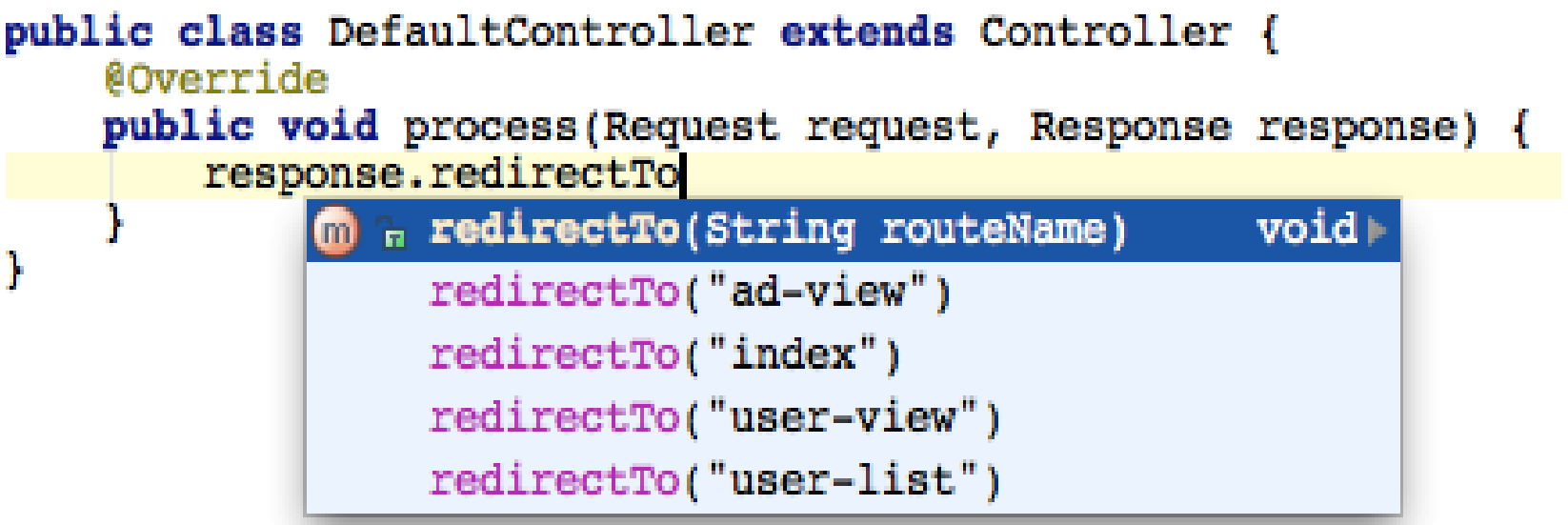
- Создать рабочее окружение проекта “сбоку” от IDE и на основе взаимодействия с ним улучшать автодополнение
  - Решает проблему метапрограммирования
  - При этом для решения 3 пункта необходимо изменить код классов

Решение

- Классы, реализующие метод List<String> getCompletionList(), способны изменять автодополнение в IDE
  - Все строчки, возвращенные методом, будут добавлены в автодополнение
- По умолчанию реализация метода getCompletionList() просматривает все доступные методы с помощью Reflection
  - Обычно решает проблему с метапрограммированием

Автодополнение для Java (1)

- Подобная функциональность также реализована для языка Java
  - Функциональность, подобная поддержанной на уровне IDE для Ruby On Rails в RubyMine



Автодополнение для Java (2)

```
public List<String> getCompletionList() {
    List<String> result = new ArrayList<>();

    for (String routeName : Config.getAppConfig().getRouter().getRoutes()) {
        result.add("redirectTo(\"" + routeName + "\")");
    }

    return result;
}
```

Результат

- Предложено и реализовано решение, улучшающее автодополнение для Groovy и для Java, в виде плагина к IDE IntelliJ IDEA
  - github.com/ttim/rcompletion
  - github.com/ttim/rcompletion-example
  - Плагин Runtime Completion в репозитории IDEA
- abishev.timur@gmail.com

Развитие

- Увеличить стабильность работы
- Отделить код для поддержки автодополнения от классов, в которых он используется
- Добавить поддержку языка Scala
- Попытаться добавить язык не из семейства основанных на jvm языков