

Сравнение исходных текстов программ на основе синтаксических деревьев

Пивоваров А. М.

научный руководитель

Лиходеев К. Н.

Кафедра математических и информационных технологий
Академический университет

2014

Сравнение программ

Цель

Выделение изменившихся участков

Проблемы

- ▶ Существенные и несущественные изменения
- ▶ Непонимание специфики сравниваемых файлов

```
public class Example {  
    public void foo1()  
    {  
        bar();  
        return;  
    }  
}  
  
public class Example {  
    public void foo1() {  
        zot1();  
        return;  
    }  
  
    public void foo2() {  
        zot2();  
        return;  
    }  
}
```

Цели и задачи

Цель

Разработать решение, позволяющее сравнивать исходные тексты программ на основе анализа их синтаксических деревьев

Задачи

- ▶ Сформулировать подход к решению задачи
- ▶ Предложить алгоритм сравнения синтаксических деревьев
- ▶ Реализовать предложенные идеи как расширение для среды разработки “IntelliJ IDEA”

Существующие решения

XML

- ▶ XMLUnit, `<oxygen/>`, DiffDog, diffxml, ...

Eclipse Modelling Framework (EMF)

- ▶ EMF Compare

Исходный код

- ▶ Smart Differencer

Предложенный подход

Требования

- ▶ Текстовое представление результата
- ▶ Детектирование перемещений
- ▶ Возможность настройки под конкретный случай

Допущения

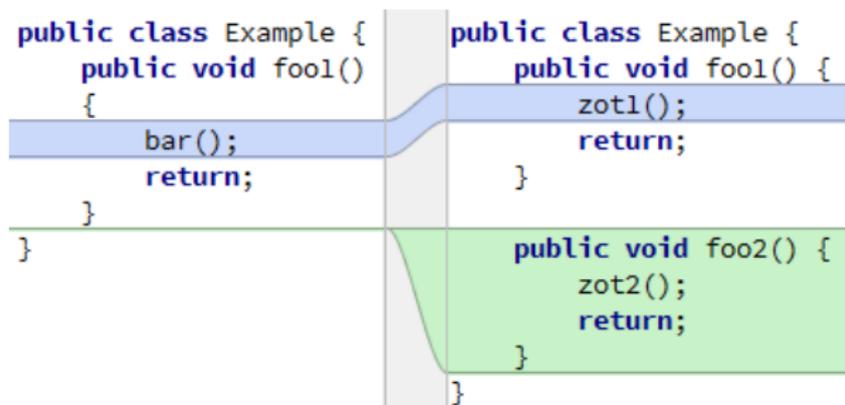
- ▶ Сравнительно низкая производительность

Результаты

Разработан полнофункциональный прототип, позволяющий сравнивать исходные тексты программ на основе их синтаксических деревьев

Преимущества

- ▶ Учет структуры языка
- ▶ Настройки
 - ▶ Игнорирование отдельных конструкций
 - ▶ Игнорирование замен и переименований



Примеры

Переименование переменной

Сверху:

Настройки по умолчанию

Снизу:

Учет переименования
локальной переменной

```
public class Example {  
    public int value;  
  
    public int foo() {  
        return value;  
    }  
  
    public int foo2() {  
        int value;  
        value = 5;  
        return value;  
    }  
}
```

```
public class Example {  
    public int value;  
  
    public int foo() {  
        return value;  
    }  
  
    public int foo2() {  
        int data;  
        value = 5;  
        return data;  
    }  
}
```

```
public class Example {  
    public int value;  
  
    public int foo() {  
        return value;  
    }  
  
    public int foo2() {  
        int value;  
        value = 5;  
        return value;  
    }  
}
```

```
public class Example {  
    public int value;  
  
    public int foo() {  
        return value;  
    }  
  
    public int foo2() {  
        int data;  
        value = 5;  
        return data;  
    }  
}
```