Генерация фрагментов исходного кода с помощью статистических методов

Кита Михаил Евгеньевич, 371 группа

Научный руководитель: ст. пр. Я. А. Кириленко

Введение

- Часто требуется быстро найти решение небольшой задачи
- На поиск тратится много времени
- Альтернатива поиску – генерация фрагмента
- Подходы разных людей к решению задач повторяются
- Следовательно, код можно предсказать
- Система, умеющая это делать, была бы полезной при разработке

Постановка задачи

- *Цель*: реализовать систему для генерации фрагментов кода на языке C# по набору вызовов API.

- Для этого нужно:
  - реализовать инструмент для сбора данных
  - собрать данные, необходимые для работы алгоритма
  - реализовать алгоритм генерации кода

Существующие решения

- Работы по генерации кода стали появляться лишь в последние годы
- Используются статистические методы
- Исследования, рассмотренные в рамках курсовой:
  - SWIM
  - T2API

Данные

- Пакеты из репозитория NuGet
  - Более 80 000 уникальных пакетов
  - Цепочки извлекались из библиотек
  - Нет проблем с неявными типами

- На текущий момент собрано
  - 4100 пакетов
  - 612000 методов

Алгоритм

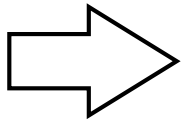
- Упорядочить входную цепочку
- Расширить информацией о контексте
- Сгенерировать код

Пример работы

```
string line = null;
Queue<string> queue = new Queue<string>();
StreamReader file = new StreamReader("file");

while ((line = file.ReadLine()) != null)
{
    queue.Enqueue(line);
}

file.Close();
```



string.new

Queue<string>.new

StreamReader.new

StreamReader.ReadLine

Queue<string>.Enqueue

StreamReader.Close

Пример работы

```
string varString = new string();
StreamReader streamReader = new StreamReader();
Queue<string> queue = new Queue<string>();

while (true)
{
    streamReader.ReadLine();

    if (...)
    {
        streamReader.Close();
    }

    queue.Enqueue(...);
}
```

Пример работы

```
string varString;
StreamReader streamReader = new StreamReader("file");
Queue<string> queue = new Queue<string>();

while (true)
{
    varString = streamReader.ReadLine();

    if (varString == null)
    {
        streamReader.Close();
        break;
    }

    queue.Enqueue(varString);
}
```

Результаты тестирования

- Точность: 58.63%
- Среднее время генерации фрагмента: 231 мс

Дальнейшая работа

- Увеличить количество собираемой информации
- Улучшить построение логических выражений для циклов и условных операторов
- Добавить подбор наиболее подходящей перегрузки метода
- Создать расширение для популярной IDE

Заключение

- Создан инструмент, позволяющий автоматически собирать данные для задач, связанных с исходным кодом
- Собран набор данных, достаточный для работы алгоритма
- Реализован алгоритм, позволяющий по цепочкам вызовов API генерировать фрагменты кода
- Промежуточные результаты представлены на конференциях