

УДК 004.4'414

Статический анализ кода, использующего Reflection (C#)

Шалин Н. А., Университет ИТМО, Санкт-Петербург

Научный руководитель: Исаев И. В., Университет ИТМО, Санкт-Петербург

## Введение

При разработке прикладного ПО часто возникает потребность в использовании сторонних библиотек и/или фреймворков. При компиляции эти зависимости вызывают увеличение размера получаемого исполняемого файла. Существующие инструменты для удаления неиспользуемого кода не могут выявлять код, используемый через API, предоставляемое библиотекой Reflection. Это приводит к тому, что в результирующих исполняемых файлах удалён код, который на самом деле должен там присутствовать, что приводит к ошибкам во время исполнения программы.

Целью данной работы является изменение статического анализатора языка C#, так, чтобы убрать ложно-положительные срабатывания предупреждений вида “method is never used” и процесса компиляции таким образом, чтобы полученные в результате исполняемые файлы были минимально необходимого размера.

## Базовые положения

Модификация стандартного статического анализатора будет осуществляться с использованием Roslyn – платформы, предоставляющей API для интеграции с процессом анализа и компиляции.

Удаление неиспользуемого кода будет осуществляться инструментом Mono.Linker с дополнительными модификациями процесса компиляции, которые так же будут реализованы с использованием Roslyn API.

Методы Reflection, такие как Invoke или CreateInstance, не регистрируются текущим анализатором, поэтому требуется добавить правила, по которым будут отслеживаться типы, передаваемые в подобные методы в качестве аргументов. Эти типы, потом следует пометить для того, чтобы Linker не удалил их.

## Итоговые результаты

Разработано расширение анализатора, которое корректно находит использование конструкторов класса при передаче его типа в метод CreateInstance.