**BC/NW 2021№ 1 (37):9.2**

**АВТОМАТИЧЕСКИЙ РЕФАКТОРИНГ ИСХОДНОГО ТЕКСТА ПРОГРАММ**

**Быков А. М., Орлов Д. А.**

Рефакторинг исходного текста программ является актуальной задачей и широко применяется для упрощения исходного текста программ, изменения структуры исходного текста программ для облегчения поддержки и развития исходного кода программы [1]. Рефакторинг исходного текста программы является важным этапом в цикле разработки программного обеспечения. Существует множество различных методов рефакторинга исходного текста программы, например, выделение новых функций, удаление повторяющихся или неработающих участков кода, и др. [1, 2]. Для автоматизации процесса рефакторинга существуют специальные программные средства, например, CodeRush, ReSharper и др. Также, поддержка автоматического рефакторинга имеется во многих средах разработки (наиболее популярные примеры — среды разработки Visual Studio, Eclipse, IntelliJ IDEA). Среди преимуществ автоматического рефакторинга исходного текста программ важно отметить такие, как скорость рефакторинга (рефакторинг, проводимый вручную, требует больше времени) и уменьшение количества ошибок, совершенных в процессе рефакторинга (в результате рефакторинга, проводимого вручную, возможны ошибки, при наличии которых возможна некорректная работа программы). К инструментам автоматического рефакторинга предъявляются довольно высокие требования, такие, как корректность преобразования исходного текста программы (результаты работы программы после рефакторинга исходного текста должны быть точно такими же, как до рефакторинга), поддержка различных методов рефакторинга. В рамках работы над магистерской диссертацией предполагается разработать экспериментальную версию программного средства, проводящую автоматизированный рефакторинг исходного текста программ. Рефакторинг будет проводиться методом выделения повторяющихся поддеревьев синтаксического дерева программы (например, повторяющиеся участки кода, функции, имеющие незначительные отличия), и преобразования данных поддеревьев в новые функции или классы. Предполагаемый язык обрабатываемых программ — C++, для проверки результатов преобразования предполагается использовать статический анализатор кода Cppcheck.

**Литература**

1. М. Фаулер., Бек К., Брант Д., Робертс Д., Апдайк У. Рефакторинг: улучшение существующего кода = Refactoring: Improving the Design of Existing Code (2000). — Спб: Символ-Плюс, 2009. — 432 с.

2. Д. Кериевски. Рефакторинг с использованием шаблонов = Refactoring to Patterns. — Вильямс, 2008. — 400 с.