

УДК 371.315.7
ББК 4421.27

ГСНТИ 14.35.07

Код ВАК 13.00.02

Коваленко Вадим Александрович,

программист, ООО «ЭргоСОЛО»; аспирант, Рязанский государственный университет им. С. А. Есенина; 129272, г. Москва, Суцёвский Вал, 64; e-mail: vadik.kovalenko@gmail.com.

**МЕТОДЫ РЕФАКТОРИНГА ПРОГРАММНОГО КОДА
В ПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКЕ СТУДЕНТОВ**

КЛЮЧЕВЫЕ СЛОВА: информационные компетенции; информационные технологии; образовательный стандарт; рефакторинг; качество программного обеспечения.

АННОТАЦИЯ. Анализируются современные проблемы разработки программного обеспечения, связанные с качеством исходного кода. Одним из перспективных направлений выхода из сложившейся ситуации представляется внедрение в учебный план дисциплины «Рефакторинг программного кода».

Kovalenko Vadim Alexandrovich,

Programmer, LLC "ErgoSOLO"; Post-graduate Student, Ryazan State University named after S. A. Esenin; Moscow, Russia.

REFACTORING CODE TECHNOLOGY IN TRAINING STUDENTS

KEY WORDS: information competencies; information technology; educational standard; refactoring; quality of software.

ABSTRACT. The article analyzes the current problems of software development related to quality of code. The author describes the current situation and perspectives of the introduction of the discipline "Refactoring code" in the curriculum of engineering specialties.

Современное отечественное образование позволяет подготавливать высококвалифицированных IT-специалистов. Однако материально-техническая база фирм-разработчиков в значительной степени отличается от вузовских компьютерных лабораторий и оснащением, и технологическими условиями, что приводит к появлению необходимости переподготовки IT-специалистов в области оценки качества программного обеспечения (ПО). В последнее десятилетие в сфере информационных технологий появилось достаточно большое количество фундаментальных и прикладных работ, в которых обосновывается необходимость использования качественной архитектуры в процессе разработки ПО.

Принципам оценки качества ПО в области программной инженерии посвящены многочисленные работы зарубежных авторов. В ряде исследований [6; 5; 7; 3] указывается на огромное количество неудачных программных продуктов, единственным недостатком которых является пренебрежение архитектурными решениями. В результате этого на рынке ПО появляются «прямолинейные системы», в которых не реализуются функциональные возможности масштабирования и адаптации информационных систем к изменяющимся коммерческим требованиям, при этом возникают серьезные трудности, связанные с затратой значительных временных и интеллектуальных ресурсов на сопровождение программных продуктов.

В стратегии развития успешных компаний в области разработки программных

продуктов и ПО можно выделить следующие наиболее общие характеристики: точное измерение продуктивности и качества программного кода; планирование и оценка разрабатываемых программных продуктов; использование наиболее эффективных методов и инструментов разработки ПО.

Эксперты в области разработки ПО применяют расширенный набор техник, позволяющих качественно улучшать структурную целостность и производительность программных продуктов. Такой подход называется рефакторингом. Наиболее точное определение рефакторинга программного кода дает в своей работе Мартин Фаулер, понимающий под рефакторингом процесс изменения программной системы, при котором происходит систематическое улучшение внутренней структуры кода без изменения его «внешнего поведения» [3; 4]. Технология рефакторинга направлена на реконструирование компонентов программы, поскольку переработать архитектуру значительно сложнее и потенциально опаснее для программной системы. На сегодняшний день выделяют два типа рефакторинга: рефакторинг программного кода и рефакторинг баз данных.

Целью проведения рефакторинга является улучшение логики и прозрачности программного кода, результатом чего является:

- улучшение читаемости программного кода;
- упрощение структуры программного кода для сопровождения ПО и его модификации;

- улучшение внутреннего качества программного продукта;
- повышение производительности и расширяемости программного продукта.

Анализ опыта зарубежных исследователей и отечественных коммерческих учебных центров, предоставляющих услуги по проведению тренингов, курсов повышения квалификации специалистов, показал, что будущие IT-специалисты должны владеть технологией и методиками разработки программных продуктов с возможностью их адаптации к постоянно изменяющимся технологическим процессам и коммерческим требованиям предприятий. Если опираться на концепцию структурного и объектно ориентированного проектирования, то программист должен быть сконцентрирован непосредственно на деталях конструкции и значительно меньше внимания уделять общей методологии разработки программного продукта [6]. Однако, как показывает практика, процесс создания качественного программного кода в значительной степени сводится к приобретению опыта в области разработки ПО. Таким образом, принципиально новое качество подготовки IT-специалистов, направленной на формирование профессиональных компетенций в области разработки программных продуктов, выбора технологий для реинжиниринга и миграции существующих информационных систем и ПО, а также ответственности за принятие эффективных и взвешенных решений, может быть обеспечено за счет экстраполяции теоретических знаний на практику. Введение дисциплины «Рефакторинг программного кода» на младших курсах обучения студентов при параллельном изучении дисциплин по программированию обеспечит комплексное формирование знаний в области разработки качественных программных продуктов и систем, отличающихся следующими особенностями: следованием стандартам программирования, поощрением связанности объектов и минимизированием взаимосвязанности объектов.

Зарубежные авторы (У. Апдайк, К. Бэк, М. Фаулер и др.) отдельно рассматривают использование техник рефакторинга для объектно ориентированного программирования (ООП). Это обусловлено наличием большого числа методов рефакторинга, направленных на преобразование внутренней структуры объектов, их иерархии наследования и взаимодействия [3]. Однако область применения методов рефакторинга не ограничивается исключительно ООП. Базовые методы рефакторинга могут применяться, например, в процедурном программировании для высокоуровневых языков программирования, таких как С, паскаль,

бейсик, фортран и др. Использование методов рефакторинга по областям применения может быть дифференцировано следующим образом:

- составление методов (composing methods);
- перемещение функций между объектами (moving features between objects);
- организация данных (organizing data);
- упрощение условных выражений (simplifying conditional expressions);
- упрощение вызовов методов (making method calls simpler);
- решение задач обобщения (dealing with generalization);
- крупные рефакторинги (big refactorings).

Такая дифференциация позволяет сформировать методологию рефакторинга для специалистов с высоким уровнем подготовки, имеющих профессиональный опыт разработки ПО. Для начинающих специалистов и студентов такая методология является сложной, поэтому классифицируем использование методов рефакторинга по областям назначения:

- рефакторинг на уровне переменных;
- рефакторинг на уровне функций;
- рефакторинг на уровне классов;
- рефакторинг алгоритмических структур.

В данной терминологии, созданной с проекцией на ООП, рефакторинг на уровне переменных подразумевает не только глобальные и локальные переменные, но и свойства классов; рефакторинг на уровне функций так же действует для процедур или методов классов (в зависимости от специфики языка программирования).

В условиях реформирования системы образования особую значимость приобретает модернизация и разработка образовательных программ подготовки бакалавров и магистров на основе федеральных государственных образовательных стандартов высшего профессионального образования на базе компетентностного подхода [2; 1].

При обучении информатике студентов и школьников продолжает доминировать парадигма процедурного программирования, не обеспечивающая в полной мере формирование профессиональных компетенций в области разработки программных продуктов. Предложенная выше классификация применения техник рефакторинга позволяет разработать структуру и содержание новой дисциплины, «Рефакторинг программного кода», для подготовки студентов IT-специальностей. Целью этого курса является формирование профессиональных компетенций в области применения методов рефакторинга наряду с получением практических навыков программи-

рования и создания программных продуктов. При этом обучение направлено не только на изучение синтаксиса и алгоритмических конструкций языков программирования, но и на качество программного кода, что позволит обеспечить грамотное проектирование и создание эффективных программных продуктов и информационных систем. Процесс написания программного кода, основанный на периодическом применении техник рефакторинга, наделяет программный продукт свойством гибкости. Гибкость в данном контексте означает способность к адаптации к изменяющимся требованиям и технологической базе, в том числе на этапе конструирования проекта.

Для формирования профессиональных компетенций у будущих IT-специалистов разработана структура программы дисциплины «Рефакторинг программного кода», которая содержит пять разделов и включает следующие темы.

1. *Введение в технологию рефакторинга.* Основные определения и термины. История развития и становления технологии рефакторинга в России и за рубежом. Базовые принципы рефакторинга. Сильные и слабые стороны рефакторинга.

2. *Основы тестирования программного кода.* Модульное тестирование ПО. Принципы разработки тестов. Автоматизированные средства для проведения тестирования программного кода.

3. *Признаки некачественного программного кода.* Дублирование данных. Избыточность данных. Архитектурные проблемы рефакторинга программных продуктов.

4. *Использование методов технологии рефакторинга по областям назначения.* Рефакторинг на уровне переменных. Рефакторинг на уровне функций. Рефакторинг на уровне классов. Рефакторинг на уровне алгоритмических структур.

5. *Специализированные инструментальные средства для автоматизации рефакторинга.* Универсальные инструменты проведения рефакторинга. Специфические инструменты проведения рефакторинга.

Отметим, что раздел «Основы тестирования программного кода» является важным звеном в дисциплине «Рефакторинг программного кода», поскольку при проведении рефакторинга программного кода предварительным условием является наличие надежных самопроверяющихся тестовых комплектов, позволяющих вовремя диагностировать ошибки и исключить их появление.

Сформулируем требования к результатам освоения дисциплины «Рефакторинг программного кода» для будущих специалистов инженерных специальностей.

После изучения дисциплины студент должен знать:

- что такое рефакторинг программного кода, каковы основные методы рефакторинга;
- признаки проблемного кода и быстрые способы поиска некачественного кода;
- что такое тестирование программного кода, каковы фундаментальные принципы тестирования;
- минимальный набор специализированных средств для автоматизации процесса рефакторинга;
- минимальный набор специализированных средств для проведения тестирования программного кода.

Студент должен иметь представление:

- о назначении, целях и причинах применения рефакторинга;
- достоинствах и недостатках рефакторинга;
- проблемах, возникающих при проведении рефакторинга;
- технических и практических критериях для инструментов рефакторинга.

Студент должен иметь практические навыки:

- внедрения рефакторинга в процесс разработки программного обеспечения;
- оперативного идентифицирования проблемного кода и устранения найденных недостатков;
- проведения эффективного анализа программного кода с целью его улучшения;
- применения различных техник рефакторинга для улучшения структурной целостности и производительности существующего программного обеспечения;
- использования специализированных средств для проведения тестирования программного кода;
- использования специализированных средств для автоматизации процесса проведения рефакторинга.

В заключение отметим, что использование методов рефакторинга носит довольно общий характер и, следовательно, дисциплина «Рефакторинг программного кода» может быть интегрирована в учебные планы различных типов образовательных учреждений вне зависимости от используемого в обучении языка программирования. Введение в вузах дисциплины «Рефакторинг программного кода» повысит качество подготовки молодых IT-специалистов, обеспечит формирование профессиональной компетентности в области разработки программных продуктов и информационных систем, повысит конкурентоспособность специалистов на рынке труда, а также

разработанного ими ПО. Среди личностных качеств, формированию которых способствует данная дисциплина, отметим перфекционизм и системное мышление, предполагающие следование установленному плану до окончательной его реализации. Приме-

нение методов рефакторинга программного кода и смежных технологических достижений в коммерческих целях позволит создавать качественные программные продукты в установленные сроки и в рамках заявленного бюджета.

ЛИТЕРАТУРА

1. Герова Н. В. Проблемы и пути совершенствования системы отечественного высшего профессионального образования // Российский научный журнал. 2010. № 14. С. 91–94.
2. Герова Н. В. Программы дисциплин макромодуля «Информатика и ИКТ» при подготовке бакалавров педагогического направления гуманитарного профиля // Российский научный журнал. 2012. № 30. С. 138–147.
3. Мартин Р. Чистый код: создание, анализ и рефакторинг. СПб. : Питер, 2011. (Библиотека программиста).
4. Мартин Фаулер. Рефакторинг. Улучшение существующего кода. СПб. : Символ-Плюс, 2013.
5. Alshayeb M. Empirical investigation of refactoring effect on software quality // Information and Software Technology. 2009. № 51. P. 1319–1326.
6. Rainwater J. H. Herding Cats: A Primer for Programmers Who Lead Programmers. Apress, 2002.
7. Mazni O. [et al.]. Educational Approach of Refactoring in Facilitating Reverse Engineering // International Journ. on Computer Science and Engineering. 2010. № 2. P. 577–581.

Статью рекомендует канд. техн. наук, доц. М. В. Лапенко.