

# **ИСПОЛЬЗОВАНИЕ АСПЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА ПРИ РЕФАКТОРИНГЕ ПРОЕКТОВ N2CMS И ORCHARD ПРИ ПОМОЩИ ASPECT.NET**

Н.В. Волков, аспирант кафедры информатики СПбГУ,  
volkov.nikita@yahoo.com

Д. А. Григорьев, доцент, к. ф.-м. н., кафедра информатики СПбГУ,  
gridmer@yandex.ru

## **Аннотация**

В данной статье описаны предпосылки к использованию аспектно-ориентированного метода для улучшения качества кода некоторых модулей крупных проектов. В статье приведен набор разработанных рефакторингов и показано их влияние на проект.

## **Введение**

Для структуризации и упрощения распределения задач в проекте, удобным средством является декомпозиция проекта на отдельные модули. Однако для некоторых видов задач характерно «сквозное» присутствие во множестве частей программы, в связи с невозможностью выделения их в одном конкретном модуле. Функциональность, реализующую такие задачи, принято называть «сквозной функциональностью». Примерами являются задачи организации прав доступа, выполнение пред- и пост- условий в модулях, обработка ошибок, регистрация действий пользователя и другие.

## **АОП-рефакторинг**

Одной из проблем, характерной для реализации сквозной функциональности в проекте, является возникновение неудобств при попытке внесения изменений, из-за необходимости вносить большое количество изменений в одни и те же участки кода. Помимо этого, реализация сквозной функциональности может усложнять восприятие другой функциональности, вплетаясь в ее код, что на практике может серьезно усложнять поддержку кода. Для решения такого рода проблем сквозной функциональности, может быть использован аспектно-

ориентированный подход (АОП), который позволяет выделять сквозную функциональность в единственный модуль (аспект).

Рефакторингом называют изменение написанного ранее кода, не влияющее на его поведение, направленное на улучшение структуры кода. АОП-рефакторинг [1] предполагает ряд аспектно-ориентированных преобразований, направленных на выделение из целевого кода сквозной функциональности в аспекты. Для реализации АОП-рефакторинга на платформе .NET был создан АОП-инструмент Aspect.NET [2]. С помощью Aspect.NET можно определять аспекты в отдельных библиотеках классов, а затем вплетать вызовы их методов в заданные места целевой сборки. Реализация аспекта в Aspect.NET осуществляется статическими методами, которые затем будут вставлены в заданные точки внедрения в сборке проекта, либо аспектным наследником, заменяющим своего родителя в целевом проекте.

АОП-рефакторинг может быть применим как в небольших, так и в крупных проектах для улучшения качества некоторых модулей, однако наибольшую эффективность этот метод показывает при использовании его в крупных проектах. Целью работы была разработка АОП-рефакторингов крупных CMS-проектов при помощи Aspect.NET. В качестве цели рефакторинга были выбраны open-source cms проекты под названием n2cms и Orchard[3], в рамках данной работы было произведено некоторое количество рефакторингов:

Для n2cms:

- Извлечение методов, выполняющих функционально схожую работу. Приведен аспект, обновляющий записанные значения в базе данных.
- Перенос обработки исключений в два аспекта, реализующих чтение событий из журнала и обработку изменения местоположения файла.
- В отдельный аспект вынесены методы, отвечающие за протоколирование вызова определенного метода.

Для Orchard:

- Перенос обработки исключений в аспект, реализующий чтение событий из журнала.
- При логировании и трассировке, методы содержат одинаковый код, который можно вынести в отдельный метод. Строки, отвечающие за журналирование, выносятся в отдельные действия.

- Функциональность методы, выполняющие форматирование различных типов данных была перемещена в аспект. Полученные действия также можно использовать вторично.
- В классе ContainerService находятся методы, которые взаимодействуют с контейнерами. Эти методы были вынесены в отдельный аспект.

Ниже приведен пример выделения в аспект функциональности протоколирования при удалении элементов:

```
public class aN2CMSClass {
    ...
    protected virtual void OnItemDeleting(ContentItem item) {
        //Здесь раньше было протоколирование
        //Прочая бизнес-логика...
    }
}
//Аспект содержится в отдельном проекте
[ReplaceBaseClass]
public class LoggingAspect {
    protected override void OnItemDeleting(ContentItem item) {
        IPrincipal user = _webContext.User;
        if (!_security.IsAuthorized(item, user)) {
            logger.InfoFormat("User {0} not authorized for {1}.",
                _webContext.User, item);
            throw new PermissionDeniedException(item, user);
        }
        //Переадресация вызова к бизнес-логике
        base.OnItemDeleting(ContentItem item);
    }
}
```

Оценка полученных результатов проводилась при помощи метрик Maintainability Index, содержащихся в Visual Studio и показывающих качество кода. В каждом из случаев были получены положительные результаты, например, индекс качества целевого класса для записи в базу данных улучшился на 20 процентов, а индексы других рефакторингов более чем на 10.

## Заключение

Таким образом, следует отметить возможность и перспективность использования АОП-рефакторингов в крупных проектах. Использование этого метода позволяет сократить расходы на тестирование и сопровождение кода, а также дает возможность реального повторного использования разработанных функциональностей.

## Литература

- [1] Григорьева А. В. *Аспектно-ориентированный рефакторинг облачных приложений MS Azure с помощью системы Aspect.NET* // *Компьютерные инструменты в образовании* // СПб. Изд-во АНО "КИО". 2012. № 1., - 21 с.
- [2] Григорьев, Д.А. *Реализация и практическое применение аспектно-ориентированной среды программирования для Microsoft .NET* // *Научно-технические ведомости* // СПб. Изд-во СПбГПУ. 2009.
- [3] Сайт проекта N2CMS // <http://n2cms.com>
- [4] Сайт проекта Orchard // <http://www.orchardproject.net>