

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РЕФАКТОРИНГ АРХИТЕКТУРЫ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Демидовой Анны Дмитриевны

Научный руководитель

Доцент

Е.В. Кудрина

Зав. кафедрой

к.ф.-м.н.

М.В. Огнева

Саратов 2018

ВВЕДЕНИЕ

Актуальность темы. Процесс проектирования архитектуры приложения – очень непростая задача. Даже опытные разработчики сталкиваются с большими трудностями при поиске лучших архитектурных решений. Роль архитектора является одной из ключевых в проектах разработки, и от выбора архитектуры зависят многие решения на разных этапах жизненного цикла проекта. Наперёд все возможные проблемы знать невозможно, поэтому часто возникает потребность в изменении архитектуры. В силу того, что задача изменения архитектуры в настоящее время актуальна, становится важно найти грамотный подход к её решению. В этом помогает рефакторинг архитектуры. В процессе рефакторинга архитектуры приложение меняется таким образом, что впоследствии вынужденное изменение архитектуры будет осуществляться легче.

Вышесказанное определило *цель бакалаврской работы* – изучить методологию архитектурного рефакторинга и реализовать её на практике.

Поставленная цель определила следующие *задачи*:

1. рассмотреть понятие «архитектура программного обеспечения» и обосновать её важность;
2. рассмотреть понятие «рефакторинг», а также основные различия классического и архитектурного рефакторинга;
3. изучить инструментальные средства, используемые для рефакторинга архитектуры;
4. реализовать рефакторинг архитектуры windows-приложения.

Методологические основы архитектуры программного обеспечения и её рефакторинга представлены в работах Л. Басса [2], Д. Спинеллиса, Г. Гусиоса [4], Р. Мартина [6], М. Фаулера, К. Бека, Д. Бранта [8].

Практическая значимость бакалаврской работы. В данной работе представлено описание инструментальных средств – ReSharper и StarUML, а также приведены примеры их использования для выполнения рефакторинга архитектуры desktop-приложений

Структура и объём работы. Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и одного приложения. Общий объем работы – 49 страниц, из них 46 страниц – основное содержание, включая 25 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы проектирования архитектуры программного обеспечения» подробно рассматривает понятие «архитектура программного обеспечения», излагает причины важности хорошей архитектуры [1-6]. Также в данном разделе рассмотрены цели архитектуры, основные принципы её проектирования и проведён обзор архитектурных стилей [1, 7].

Раздел содержит два подраздела.

«Основные понятия и определения» – в этом подразделе проводится сравнение определений архитектуры программного обеспечения (ПО) из разных источников, обосновывается важность архитектуры ПО, описываются цели и основные принципы проектирования архитектуры ПО (принцип разделение функций, принцип единственности ответственности, принцип минимального знания, принцип «Не повторяться!», принцип минимизация проектирования наперед).

В рамках данной работы было принято решение придерживаться следующего определения понятия архитектура ПО:

Архитектура программного обеспечения (ПО) – это организация программной системы, которая описывает связи между структурными элементами и их интерфейсами, а также внешней средой; поведение, обеспечиваемое совместной работой компонентов системы и их организация в подсистемы. К архитектуре относится лишь внешняя часть компонентов системы, а не то, что связано с внутренней реализацией.

Подраздел «Обзор архитектурных стилей» содержит определение термина архитектурный стиль [1], а также описание следующих архитектурных стилей: «Архитектура клиент/сервер», «Компонентная архитектуры», «Проектирование на основе предметной области (Domain Driven Design, DDD)», «Многослойная архитектура», «N-уровневая / 3-уровневая архитектура», «Объектно-ориентированная архитектура». Описываются их отличительные особенности, основные достоинства и недостатки [7].

На основе анализа архитектурных стилей делается вывод, что на проектирование и создание приложений влияет огромное количество нюансов, поэтому нельзя выделить какой-либо универсальный архитектурный стиль, подходящий для приложений любого типа. Как правило, при проектировании приложений архитектурные стили комбинируются.

Второй раздел «Теоретические основы рефакторинга архитектуры» посвящен изучению понятия «рефакторинг», сравнению архитектурного и классического рефакторинга, описанию связи рефакторинга и тестирования, изучению основных подходов и принципов рефакторинга архитектуры [8-13]. Также в разделе рассмотрены основные инструментальные средства, являющиеся вспомогательными при проведении рефакторинга архитектуры [14-19].

Раздел содержит два подраздела.

Подраздел «Место и роль рефакторинга в эволюции программного обеспечения» содержит в себе два определения рефакторинга, рассмотрение основных различий рефакторинга архитектуры и классического рефакторинга.

Также в подразделе обусловлена важность тестирования при проведении рефакторинга и рассмотрены основные подходы и принципы рефакторинга архитектуры, которые включают в себя выполнение принципов проектирования SOLID (принцип единственной ответственности, принцип

открытости/закрытости, принцип подстановки Лисков, принцип разделения интерфейса, принцип инверсии зависимостей), и основной метод рефакторинга архитектуры – метод «выделения слоев».

Подраздел «Инструментальные средства для проведения рефакторинга архитектуры» содержит в себе описание таких инструментальных средств, как ReSharper – плагин, созданный для повышения продуктивности работы в Microsoft Visual Studio и StarUML – инструмент для UML-моделирования. Рассмотрены основные возможности ReSharper, с помощью которых можно избавиться от архитектурных недостатков, таких, как нарушение принципов SOLID. Инструмент StarUML рассмотрен в качестве инструмента, позволяющего представить модель системы в графическом виде и применить метод «выделения слоев».

Третий раздел «Реализация рефакторинга архитектуры приложения «Floating Horizons» посвящен изучению приложения «Floating Horizons», использующего алгоритм плавающего горизонта для построения поверхностей вида $F(x, y, z) = 0$, выявлению его архитектурных недостатков, а также реализации рефакторинга архитектуры этого приложения с помощью инструментальных средств и методов, рассмотренных в разделе II.

Раздел содержит три подраздела.

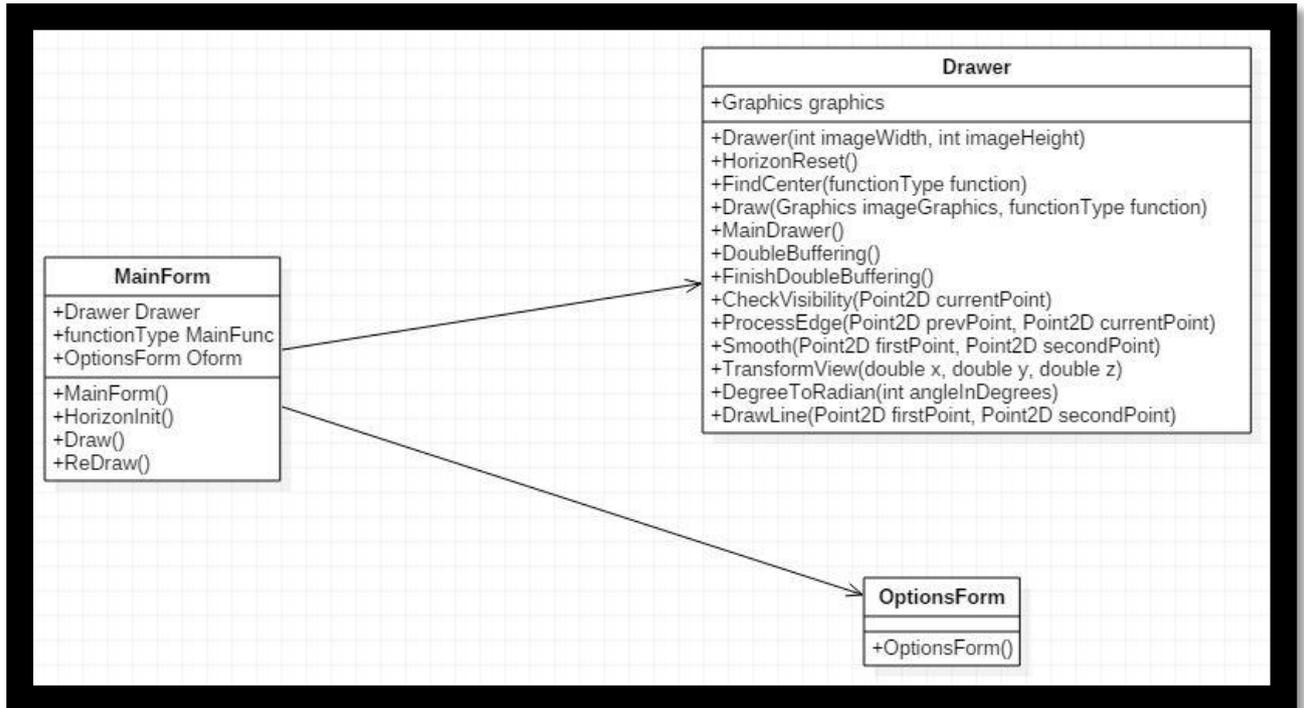
Подраздел «Постановка задачи» описывает задачу по рефакторингу архитектуры, которая перед нами стоит, а также шаги, которые предстоит совершить для реализации этой задачи.

«Планирование и реализация рефакторинга архитектуры» – подраздел содержит описание исходного приложения [20], выявление недостатков архитектуры этого приложения, а также описание шагов рефакторинга с помощью инструментальных средств и «метода выделения слоев».

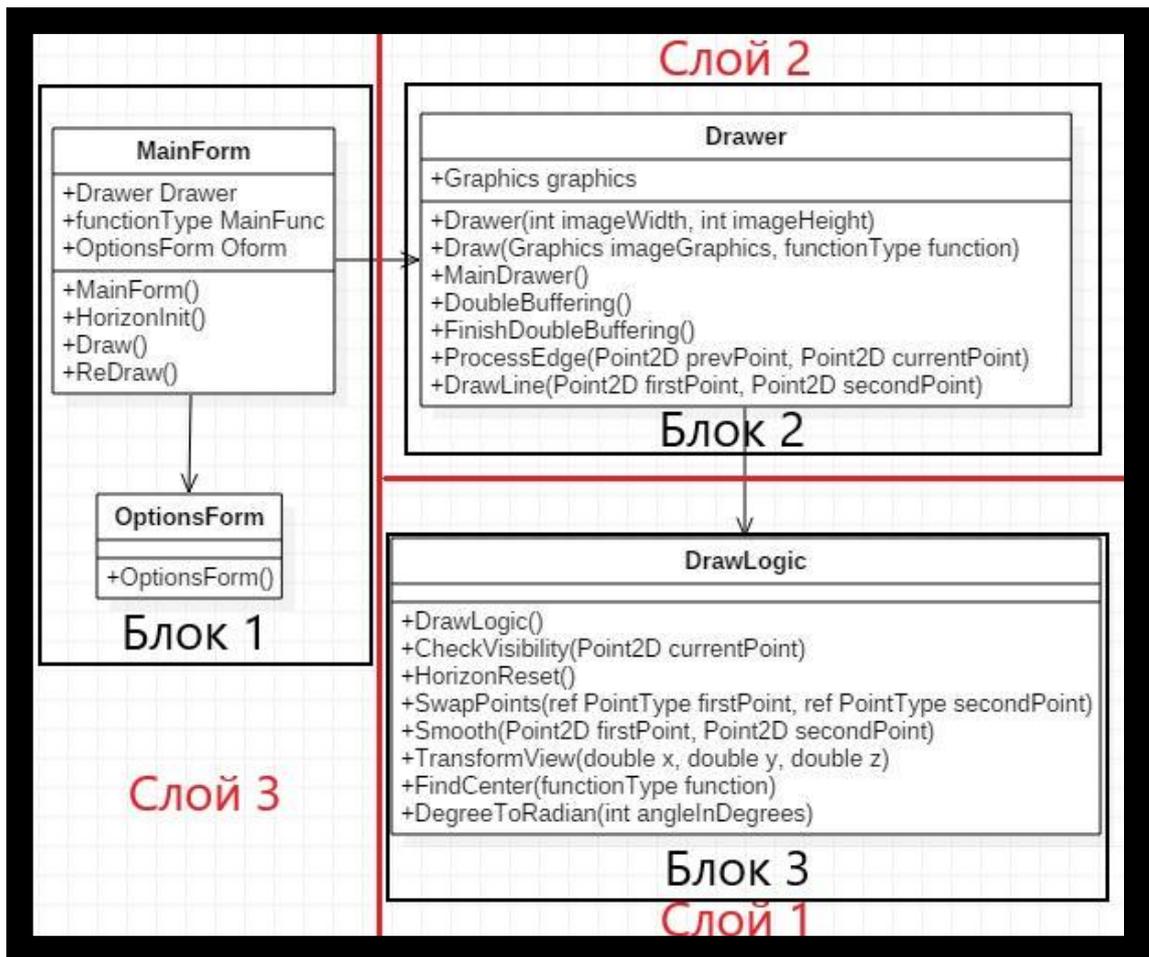
«Анализ выполненного рефакторинга архитектуры» – подраздел подводит итоги проделанной работы и обосновывает важность проделанных изменений.

Исходное приложение имело монолитную структуру, без четко

ВЫДЕЛЕННЫХ СЛОЕВ.



Итоговое приложение имеет 3-слойную архитектуру.



ЗАКЛЮЧЕНИЕ

Тема рефакторинга архитектуры на данный момент исследована не очень хорошо, но знания и умения по этой теме становятся всё более востребованы.

В ходе работы были выполнены все поставленные задачи. Рассмотрено понятие «архитектура программного обеспечения» и обоснована её важность; изучено понятие «рефакторинг», а также основные различия классического и архитектурного рефакторинга; изучены инструментальные средства, используемые для рефакторинга архитектуры; реализован рефакторинг архитектуры windows-приложения. Благодаря выполнению всех задач достигнута цель бакалаврской работы – методология архитектурного рефакторинга изучена и реализована на практике.

Проделанная работа по изучению теоретических сведений и инструментальных средств позволяет в дальнейшем без труда проводить рефакторинг архитектуры более сложных windows-приложений. Выполненный в практической части рефакторинг сделал архитектуру приложения более грамотной, вследствие чего в структуре системы стало легче разобраться. Также проведённый рефакторинг упростил возможную дальнейшую модификацию приложения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство Microsoft по проектированию архитектуры приложений [Электронный ресурс]. URL: http://download.microsoft.com/documents/rus/msdn/ры_приложений_полная_книга.pdf (дата обращения: 04.06.18). Загл. с экрана. Яз. рус.
2. Басс, Л. Архитектура программного обеспечения на практике [Текст] / Л. Басс, П. Клементс, Р. Кацман; пер. с англ. – СПб.: Питер, 2006. – 575 с.

3. Архитектурные особенности проектирования и разработки Веб-приложений [Электронный ресурс]// Библиотека on-line – URL: <https://www.intuit.ru/studies/courses/611/467/lecture/28784?page=1> (дата обращения 04.06.2018).
4. Спинеллис, Д. Идеальная архитектура. Ведущие специалисты о красоте программных архитектур [Текст] / Д. Спинеллис, Г. Гусиос; пер. с англ. Е. Матвеевой, науч. ред. А. Брагина. – СПб.: Символ-Плюс, 2010. – 528 с.
5. Розенфельд, Л. Информационная архитектура в Интернете [Текст] / Л. Розенфельд, П. Морвиль; пер. с англ. С. Маккавеевой. – СПб.: Символ-Плюс, 2005. – 544 с.
6. Мартин, Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста [Текст] / Р. Мартин; пер. с англ. Е. Матвеева. – СПб.: Питер, 2010. – 464 с.
7. Фаулер, М. Архитектура корпоративных программных приложений. [Текст] / М. Фаулер; пер. с англ. – М.: Вильямс, 2006. – 544 с.
8. Рефакторинг. Улучшение существующего кода [Текст] / М. Фаулер, К. Бек, Д. Брант и др.; пер. с англ. С. Маккавеевой, науч. ред. Е. Шпика. – СПб.: Символ-Плюс, 2003. – 432 с.
9. Рефакторинг архитектуры программного обеспечения: выделение слоев [Электронный ресурс]// Библиотека on-line – URL: <http://citforum.ru/SE/project/refactor/> (дата обращения 29.05.2018).
10. What is Architectural Refactoring? [Электронный ресурс]// Lattix – URL: <http://lattix.com/blog/2017/04/14/architectural-refactoring> (дата обращения 29.05.2018).
11. Принципы SOLID в C# [Электронный ресурс]// net & web programming – URL: <https://professorweb.ru/my/it/blog/net/solid.php> (дата обращения 29.05.2018).

- 12.Шпаргалка по SOLID-принципам с примерами на PHP [Электронный ресурс] – URL: <https://habr.com/post/208442/> (дата обращения 29.05.2018).
- 13.Назаров, С.В. Архитектурный рефакторинг программных систем [Текст] / С.В. Назаров. – Москва: Высшая школа экономики, 2014.
- 14.ReSharper [Электронный ресурс]// Свободная энциклопедия – URL: <https://ru.wikipedia.org/wiki/ReSharper> (дата обращения 29.05.2018).
- 15.Extract Class refactoring [Электронный ресурс]// ReSharper – URL: https://www.jetbrains.com/help/resharper/Refactorings_Extract_Class.html (дата обращения 29.05.2018).
- 16.Extract Interface refactoring [Электронный ресурс]// ReSharper – URL: https://www.jetbrains.com/help/resharper/Refactorings_Extract_Interface.html (дата обращения 29.05.2018).
- 17.Extract Method refactoring [Электронный ресурс]// ReSharper – URL: https://www.jetbrains.com/help/resharper/Refactorings_Extract_Method.html (дата обращения 29.05.2018).
- 18.Диаграммы классов UML [Электронный ресурс]// Блог программиста – URL: <https://pro-prof.com/archives/3212> (дата обращения 29.05.2018).
- 19.Способы применения UML при разработке ПО [Электронный ресурс] – URL: <https://lektsii.org/3-98425.html> (дата обращения 04.06.2018).
- 20.Язык программирования C# [Текст] / А. Хейлсберг, М. Торгерсен, С. Вилтамут и др.; пер. с англ. – СПб.: Питер, 2012. – 784 с.