

Содержание

Об авторе	13
Благодарности	13
Введение	14
Для кого написана эта книга	15
Какие темы охватывает эта книга	16
Как организована эта книга	17
Что необходимо для использования этой книги	18
Соглашения	19
Врезки “Запах”, “Рефакторинг” и “Принцип объектно-ориентированного проектирования”	19
Врезка “Запах”	19
Врезка “Рефакторинг”	20
Врезка “Принцип объектно-ориентированного проектирования”	20
Указатель запахов	20
Указатель рефакторингов	21
Указатель принципов объектно-ориентированного проектирования	23
Исходный код	23
От издательства	23
Глава 1. Рефакторинг: о чем весь этот шум?	25
Краткий обзор рефакторинга	25
Процесс рефакторинга	26
Взгляд на ситуацию в программном обеспечении	27
Более пристальный взгляд на процесс рефакторинга	30
Код с душком	30
Трансформация кода	30
Автоматизация трансформаций рефакторинга	31
Выгоды, которые дает рефакторинг	31
Разоблачение распространенных заблуждений	35
Ни один программист не одинок	43
Рефакторинг для представителей бизнеса	45
Язык C# и рефакторинг	46
Резюме	47
Глава 2. Первая проба рефакторинга	49
Пример приложения: калькулятор калорий	49
Приложение “Калькулятор калорий” с вычислением ежедневного рекомендуемого объема калорий	50
Расширение требований: вычисление идеального веса	53
Требования растут: нужно хранить данные о пациентах	55
Рефакторинг в действии	57
Декомпозиция метода btnCalculate_Click	58
Сегмент вычисления и отображения расхождения с идеальным весом	59
Вычисление калорий и идеального веса в соответствии с полом	59

Метод <code>btnCalculate_Click</code> после выделения методов	60
Исследование новых классов	61
Сужение интерфейса класса <code>Patient</code>	63
Реструктуризация метода <code>DistanceFromIdealWeight</code>	67
Добавление в <code>CaloriesCalculator</code> использования класса <code>Patient</code>	68
Создание иерархии классов <code>Patient</code>	69
Реализации функциональности постоянства	74
Сохранение данных	74
Реализация функциональности отображения истории	83
Приложение “Калькулятор калорий”, улучшенная версия	86
Резюме	88
Глава 3. Сборка инструментария для рефакторинга	91
Использование автоматизированного инструмента рефакторинга	92
ReSharper от JetBrains	92
Refactor! Pro от Developer Express	93
Refactor! for ASP от Developer Express	94
Средства рефакторинга Visual Studio	95
Основы тестирования модулей: тестовая оснастка	97
Назначение каркаса тестирования модулей	97
Первое знакомство с NUnit	100
Установка NUnit	100
Работа с примерами	101
Реализация первого теста	103
Управляемый тестами подход	110
Другие инструменты тестирования	112
Управление версиями	113
Управление версиями как система резервного копирования	113
Управление версиями и параллелизм	114
Резюме	114
Глава 4. Прототип приложения: прокат автомобилей	117
Интервью с клиентом	118
Интервью с менеджером	118
Интервью с секретарем приемной	119
Интервью с сотрудником паркинга	120
Интервью с техническим персоналом	120
Первые шаги в проекте “Прокат автомобилей”	121
Актеры и сценарии использования	121
Состояния автомобиля	123
Первый набросок главного окна приложения	124
Совещание команды Rent-a-Wheels	126
Запуск прототипа в работу	126
Рассмотрение модели базы данных	126
Рассмотрение кода C#	128
Быстрый и яростный подход к программированию	132
Дизайн, управляемый базой данных	132
Приложения, основанные на графическом интерфейсе пользователя	133
Программирование, управляемое событиями	133

Быстрая разработка приложений	134
Копирование и вставка как механизм повторного использования кода	134
От прототипа к поставке готового продукта через процесс рефакторинга	135
Резюме	135
Глава 5. Основы гигиены	137
Исключение мертвого кода	137
Типы мертвого кода	139
Обычные источники мертвого кода	140
Сокращение области видимости и уровня доступа излишне открытых элементов	143
Область видимости и уровень доступа	145
Обычные источники излишней открытости	147
Как справиться с излишней открытостью	150
Использование явного импорта	151
Раздел <code>using</code> отражает зависимости в системе	151
Удаление неиспользуемых ссылок на сборки	154
Базовая гигиена проекта “Прокат автомобилей”	155
Резюме	155
Глава 6. От предметной области к коду: заполнение пробела	157
Понимание предметной области	158
Шаг 1: сбор информации	159
Шаг 2: согласование словаря	160
Шаг 3: описание взаимодействий	161
Шаг 4: построение прототипа	161
Руководства по именованию	162
Стили применения заглавных букв	163
Простые руководства по именованию	163
Хорошая выразительность: выбор правильных слов	165
Рефакторинг “Переименование” (Rename)	169
Рефакторинг “Переименование” (Rename) в Visual Studio	171
Опубликованные и общедоступные интерфейсы	172
Сравнение автономных приложений и многократно используемых модулей	173
Модификация опубликованных интерфейсов	177
Рефакторинги “Переименование” и “Безопасное переименование” и приложение “Прокат автомобилей”	181
Резюме	182
Глава 7. Выделение метода, как мера против дублированного кода	183
Инкапсуляция кода и сокрытие деталей	183
Соккрытие информации и реализации	184
Декомпозиция методов	187
Пример длинного метода: вычисление длины окружности	187
Выделение кода вычисления длины окружности	189
Выделение кода вычисления радиуса	193
Выделение кода “Ожидание закрытия пользователем”	193
Выделение кода чтения координат	194
Рефакторинг “Выделение метода” (Extract Method) в Visual Studio	197
Встраивание методов	200

Запах “Дублирование кода” (Duplicated Code)	202
Источники дублированного кода	203
Программирование копированием-вставкой	203
Магические литералы	204
Рефакторинг “Выделение метода” и “Замена магического литерала константой” в приложении “Прокат автомобилей”	206
Резюме	207
Глава 8. Приемы консолидации и выделения методов	209
Обработка временных переменных	209
Рефакторинг “Перемещение объявления к месту ссылки” (Move Declaration Near Reference)	210
Рефакторинг “Перемещение инициализации к месту объявления” (Move Initialization to Declaration)	213
Рефакторинг “Расщепление временной переменной” (Split Temporary Variable)	214
Рефакторинг “Встраивание временной переменной” (Inline Temp)	218
Рефакторинг “Замена временной переменной вызовом метода” (Replace Temp with Query)	220
Введение поясняющей временной переменной	223
Работа с длинными и вложенными условными операторами	224
Реорганизация методов и приложение “Прокат автомобилей”	227
Удаление дублированного кода из приложения “Прокат автомобилей”	228
Магические литералы, комментарии и слепая обработка событий в приложении “Прокат автомобилей”	232
Резюме	236
Глава 9. Обнаружение объектов	239
Краткий обзор объектно-ориентированного программирования	240
Объекты в ООП	240
Инкапсуляция и объекты	240
Рефакторинг “Инкапсуляция поля” (Encapsulate Field) в Visual Studio	242
Сохранение состояния объекта	243
Классы	244
Идентичность объекта	245
Объекты как базовые строительные блоки	246
Корневой тип Object	246
Время жизни объекта и сборка мусора	247
Проектирование классов	249
Использование артефактов анализа	249
Классы – существительные, операции – глаголы	252
Классы, ответственности и кооперации	257
Применение карточек в сеансах мозгового штурма	261
Сущности и отношения	265
Выявление скрытых классов	266
Работа с проектом, управляемым базой данных	267
Переход от процедурного к объектно-ориентированному проектированию	270
Разделение предметной области, презентации и постоянного хранения	276
Обнаружение объектов в приложении “Прокат автомобилей”	282
Резюме	289

Глава 10. Расширенные объектно-ориентированные концепции и связанный с ними рефакторинг	291
Наследование, полиморфизм и обобщения	292
Наследование	292
Сравнение наследования классов и наследования интерфейсов	295
Полиморфизм	296
Обобщения	299
Злоупотребление наследованием и решения рефакторинга	301
Ошибочное применение наследования вместо композиции и прочие злоупотребления	305
Рефакторинг наследования на примере системы печати	310
Использование делегирования вместо наследования в системе печати	317
Использование обобщений	328
Наследование и обобщенные типы в приложении “Прокат автомобилей”	331
Выделение родительского класса	332
Использование обобщений	332
Выделение класса <code>DataObjectsProvider</code>	333
Резюме	337
Глава 11. Крупномасштабная организация кода	339
Пространства имен	339
Руководства по именованию и организации пространств имен	340
Вложенные пространства имен	340
Изменение названия пространства имен по умолчанию	340
Применение директив <code>using</code>	341
Сборки	343
Бинарное повторное использование	343
Принципы организации пространств имен	345
Размышления о зависимостях	350
Организация файловой структуры проекта <code>C#</code>	357
Частичные классы	358
Организация пространств имен и наследование форм <code>Windows</code> в приложении “Прокат автомобилей”	364
Выделение родительской административной формы с использованием шаблона абстрактного вспомогательного класса	364
Реорганизация пространства имен и сборки	371
Резюме	373
Глава 12. Рефакторинг по шаблонам проектирования	375
Шаблоны проектирования: о чем весь этот шум?	376
Определение шаблонов проектирования	376
Классификация шаблонов	377
Элементы шаблонов	378
Оценка преимуществ шаблонов проектирования	378
Использование шаблонов	379
Пример шаблона проектирования: <code>Abstract Factory</code> (Абстрактная фабрика)	379
Использование шаблона <code>Abstract Factory</code>	379
Подъем объявления объекта	382

Решение	389
Последствия	392
Шаблон Dependency Injection (Внедрение зависимости)	394
Проблема при использовании шаблона Dependency Injection (DI)	394
Решение	397
Сравнение внедрения на основе конструктора и внедрения на основе свойства	397
Какую реализацию службы внедрять	398
Преимущества шаблона DI	400
Рефакторинг для использования DI	402
Рефакторинг по шаблонам и приложение “Прокат автомобилей”	403
Исключение кода, дублирующего функциональность .NET Framework	403
Внедрение классов данных в классы графического интерфейса пользователя через шаблон Dependency Injection (DI)	403
Шаблон постоянного хранения CRUD	407
Резюме	407
Глава 13. LINQ и другие расширения C# 3.0	409
Выведение типа локальных переменных	409
Автореализованные свойства	411
Расширяющие методы	413
Инициализаторы объектов, массивов и коллекций	419
Опрос объектов с LINQ	420
Старый пример в новом виде	424
Объектно-реляционное отображение с помощью LINQ to SQL	430
LINQ и приложение “Прокат автомобилей”	435
Резюме	442
Глава 14. Краткая история Веб для разработчиков и инструментарий ASP.NET	445
Refactor! for ASP.NET	446
Вызов Refactor! for ASP.NET	447
Пользовательский интерфейс Refactor! for ASP.NET	449
История и наследие HTML	453
История HTML	454
В ногу с Веб	462
Visual Studio и XHTML	462
XML и кодировка	463
Проверка DTD-определения для HTML в Visual Studio	464
Передача строгого XHTML	465
Резюме	467
Глава 15. Рефакторинг приложений ASP.NET	469
Рефакторинг HTML	470
Хорошо оформленные документы XHTML	470
Действительность XHTML	472
Инструментальная поддержка обновления унаследованной, не совместимой с XHTML разметки	474
Улучшение внешнего вида кода документов HTML	475
Отделение структуры от презентации	476
Использование HTTP с REST	482

12 Содержание

Рефакторинг кода ASP.NET	487
Модель кода ASP.NET: однофайловая или отделенного кода	487
Мастер-страницы	491
Пользовательские веб-элементы и специальные серверные элементы управления	494
Приложение “Прокат автомобилей” и рефакторинги ASP.NET	500
Резюме	503
Приложение А. Внутренняя структура прототипа приложения “Прокат автомобилей”	504
Код обработки щелчка на кнопке Hand Over	504
Код обработки щелчка на кнопке Receive	505
Код обработки щелчка на кнопке Charge	505
Код обработки щелчка на кнопке Change Branch	506
Кнопки To Maintenance и From Maintenance	510
Форма администрирования парка автомобилей	510
Процедура обработки события щелчка на кнопке Delete	511
Процедура обработки события щелчка на кнопке New	511
Процедура обработки события щелчка на кнопке Reload	511
Процедура обработки события Form Load	512
Код класса формы администрирования парка автомобилей: поля	514
Процедура обработки события щелчка на кнопке навигации	515
Процедура обработки события щелчка на кнопке Save	516
Процедура обработки кнопки Display	517
Резюме	522
Приложение Б. Настройка Refactor! for ASP.NET	523
Предметный указатель	525