

Адаптивный КОД

Гибкое кодирование с помощью
паттернов проектирования
и принципов SOLID

2-е издание

Лучшие практики



Гэри Маклин Холл

Оглавление

| | |
|--|-----|
| Введение | 14 |
| Часть I. Инфраструктуры гибкой разработки | 23 |
| Глава 1. Введение в Scrum | 25 |
| Глава 2. Введение в канбан | 73 |
| Часть II. Основы адаптивного кода | 97 |
| Глава 3. Зависимости и разделение на уровни | 99 |
| Глава 4. Интерфейсы и паттерны проектирования | 148 |
| Глава 5. Тестирование | 180 |
| Глава 6. Рефакторинг | 225 |
| Часть III. Код SOLID | 251 |
| Глава 7. Принцип единственной обязанности | 253 |
| Глава 8. Принцип открытости/закрытости | 287 |
| Глава 9. Принцип подстановки Лисков | 299 |
| Глава 10. Разделение интерфейса | 333 |
| Глава 11. Инверсия зависимостей | 363 |
| Часть IV. Применение адаптивного кода | 383 |
| Глава 12. Внедрение зависимостей | 384 |
| Глава 13. Связанность, сцепление и соразвитие | 427 |
| Приложение А. Адаптивные инструменты | 437 |
| Предметный указатель | 443 |

Содержание

| | |
|--|-----------|
| Об авторе | 13 |
| Благодарности | 13 |
| Введение | 14 |
| Кто должен читать эту книгу | 15 |
| Эта книга может быть не для вас, если... | 16 |
| Организация этой книги | 16 |
| Соглашения, принятые в этой книге | 20 |
| Системные требования | 21 |
| Загрузка кода примеров | 21 |
| Ждем ваших отзывов! | 22 |
| Часть I. Инфраструктуры гибкой разработки | 23 |
| Глава 1. Введение в Scrum | 25 |
| Scrum в сравнении с каскадным подходом | 29 |
| Роли и обязанности | 31 |
| Владелец продукта | 31 |
| Scrum-мастер | 32 |
| Команда разработчиков | 33 |
| Артефакты | 34 |
| Доска Scrum | 35 |
| Диаграммы и метрики | 50 |
| Беклоги | 55 |
| Спринт | 57 |
| Планирование выпуска | 58 |
| Планирование спринта | 59 |
| Ежедневные совещания Scrum | 61 |
| Демонстрация спринта | 63 |
| Ретроспектива спринта | 64 |
| Календарь Scrum | 66 |
| Гибкая разработка в реальном мире | 67 |
| Жесткость | 67 |
| Неспособность к тестированию | 69 |
| Метрики | 70 |
| Заключение | 72 |
| Глава 2. Введение в канбан | 73 |
| Начало работы с канбан | 74 |
| Излучатель информации | 74 |
| Ограничение выполняющихся работ | 79 |
| Защита против изменений | 79 |
| Определение законченности | 80 |

| | |
|--|------------|
| Церемониал, управляемый событиями | 81 |
| Классы обслуживания | 83 |
| Соглашения об уровне обслуживания | 83 |
| Лимиты WIP классов обслуживания | 85 |
| Люди как класс обслуживания | 86 |
| Анализ | 87 |
| Время выполнения и время цикла | 87 |
| Совокупные диаграммы последовательности действий | 90 |
| Заключение | 96 |
| Часть II. Основы адаптивного кода | 97 |
| Глава 3. Зависимости и разделение на уровни | 99 |
| Зависимости | 100 |
| Простой пример | 101 |
| Зависимости от инфраструктуры | 105 |
| Сторонние зависимости | 107 |
| Моделирование зависимостей в ориентированном графе | 108 |
| Управление зависимостями | 112 |
| Реализации или интерфейсы | 113 |
| Ключевое слово new как признак плохого кода | 113 |
| Альтернативы конструированию объектов | 117 |
| Распознавание зависимостей | 119 |
| Управление зависимостями с помощью NuGet | 131 |
| Разделение на уровни | 135 |
| Общие паттерны разделения на уровни | 137 |
| Сквозные обязанности | 143 |
| Асимметричное разделение на уровни | 144 |
| Заключение | 147 |
| Глава 4. Интерфейсы и паттерны проектирования | 148 |
| Что такое интерфейс? | 149 |
| Синтаксис | 149 |
| Явная реализация | 152 |
| Полиморфизм | 156 |
| Адаптивные паттерны проектирования | 157 |
| Паттерн “Null-объект” | 158 |
| Паттерн “Адаптер” | 163 |
| Паттерн “Стратегия” | 166 |
| Дополнительная разносторонность | 168 |
| Утиная типизация | 168 |
| Примеси | 173 |
| Текущие интерфейсы | 178 |
| Заключение | 179 |

| | |
|---|-----|
| Глава 5. Тестирование | 180 |
| Модульное тестирование | 181 |
| Организация, действие, утверждение | 181 |
| Разработка через тестирование | 186 |
| Более сложные тесты | 192 |
| Паттерны модульного тестирования | 209 |
| Написание удобных для сопровождения тестов | 209 |
| Паттерн “Строитель” для тестов | 211 |
| Паттерн “Строитель” | 211 |
| Прояснение намерения модульных тестов | 212 |
| Написание сначала тестов | 215 |
| Что такое TDD? | 215 |
| Проектирование через тестирование | 216 |
| Разработка в стиле “сначала тесты” | 217 |
| Добавочное тестирование | 218 |
| Пирамида тестирования | 218 |
| Антипаттерны для пирамиды тестирования | 219 |
| Квадранты тестирования | 220 |
| Тестирование для профилактики и устранения проблем | 222 |
| Как уменьшить MTTR? | 223 |
| Заключение | 224 |
| Глава 6. Рефакторинг | 225 |
| Введение в рефакторинг | 225 |
| Изменение существующего кода | 226 |
| Новый тип счета | 236 |
| Энергичный рефакторинг | 241 |
| Красный, зеленый, рефакторинг... перепроектирование | 241 |
| Превращение унаследованного кода в адаптивный | 242 |
| Прием золотого мастера | 243 |
| Заключение | 250 |
| Часть III. Код SOLID | 251 |
| Глава 7. Принцип единственной обязанности | 253 |
| Формулировка проблемы | 254 |
| Рефакторинг ради ясности | 257 |
| Рефакторинг для абстрагирования | 261 |
| SRP и паттерн “Декоратор” | 269 |
| Паттерн “Компоновщик” | 270 |
| Предикатные декораторы | 273 |
| Разветвляющие декораторы | 276 |
| Ленивые декораторы | 278 |
| Регистрирующие декораторы | 279 |

| | |
|--|------------|
| Профилирующие декораторы | 280 |
| Декорирование свойств и событий | 284 |
| Заключение | 286 |
| Глава 8. Принцип открытости/закрытости | 287 |
| Введение в принцип открытости/закрытости | 287 |
| Определение Мейера | 288 |
| Определение Мартина | 288 |
| Исправления дефектов | 289 |
| Осведомленность клиентов | 289 |
| Точки расширения | 290 |
| Код без точек расширения | 290 |
| Виртуальные методы | 291 |
| Абстрактные методы | 291 |
| Наследование интерфейса | 292 |
| “Проектируйте наследование или запретите его” | 293 |
| Устойчивость к изменениям | 294 |
| Предсказуемые изменения | 295 |
| Стабильный интерфейс | 295 |
| Только достаточная степень адаптации | 295 |
| Предсказуемые изменения или гипотетическое обобщение | 296 |
| Нужно ли так много интерфейсов? | 297 |
| Заключение | 298 |
| Глава 9. Принцип подстановки Лисков | 299 |
| Введение в принцип подстановки Лисков | 299 |
| Формальное определение | 300 |
| Правила LSP | 300 |
| Контракты | 301 |
| Предусловия | 303 |
| Постусловия | 304 |
| Инварианты данных | 305 |
| Правила контрактов Лисков | 307 |
| Контракты кода | 314 |
| Ковариантность и контравариантность | 322 |
| Определения | 322 |
| Правила системы типов Лисков | 329 |
| Заключение | 332 |
| Глава 10. Разделение интерфейса | 333 |
| Пример разделения | 334 |
| Простой интерфейс CRUD | 334 |
| Кеширование | 339 |
| Декорирование множества интерфейсов | 343 |

| | |
|--|------------|
| Построение клиентов | 346 |
| Множество реализаций, множество экземпляров | 346 |
| Единственная реализация, единственный экземпляр | 348 |
| Антипаттерн “Каша из интерфейсов” | 350 |
| Разделение интерфейсов | 350 |
| Потребность клиента | 351 |
| Архитектурная потребность | 357 |
| Интерфейсы с единственным методом | 361 |
| Заключение | 362 |
| Глава 11. Инверсия зависимостей | 363 |
| Структурирование зависимостей | 363 |
| Антипаттерн “Антураж” | 364 |
| Паттерн “Лестница” | 366 |
| Пример проектирования абстракций | 367 |
| Абстракции | 368 |
| Конкретные реализации | 369 |
| Абстрагирование возможностей | 372 |
| Улучшенный клиент | 378 |
| Абстрагирование запросов | 380 |
| Дальнейшее абстрагирование | 382 |
| Заключение | 382 |
| Часть IV. Применение адаптивного кода | 383 |
| Глава 12. Внедрение зависимостей | 384 |
| Непритязательное начало | 385 |
| Приложение “Список задач” | 388 |
| Построение объектного графа | 391 |
| За рамками простого внедрения | 409 |
| Антипаттерн “Локатор служб” | 410 |
| Незаконное внедрение | 413 |
| Корень композиции | 415 |
| Соглашение по конфигурации | 421 |
| Заключение | 425 |
| Глава 13. Связанность, сцепление и соразвитие | 427 |
| Связанность и сцепление | 427 |
| Связанность | 428 |
| Сцепление | 428 |
| Соразвитие | 429 |
| Имя | 430 |
| Тип | 431 |
| Смысл | 432 |
| Алгоритм | 432 |

| | |
|---|-----|
| Позиция | 433 |
| Порядок выполнения | 434 |
| Синхронизация | 434 |
| Значение | 434 |
| Идентичность | 434 |
| Измерение соразвития | 434 |
| Местонахождение | 435 |
| Неофициальное соразвитие | 435 |
| Статическое или динамическое соразвитие | 436 |
| Заключение | 436 |
| Приложение А. Адаптивные инструменты | 437 |
| Управление исходным кодом с помощью Git | 437 |
| Копирование хранилища | 440 |
| Переключение на другую ветвь | 440 |
| Непрерывная интеграция | 441 |
| Предметный указатель | 443 |