



БИБЛИОТЕКА ПРОГРАММИСТА



Сэнди Метц

Ruby

**Объектно-
ориентированное
проектирование**

 ПИТЕР®



БИБЛИОТЕКА ПРОГРАММИСТА

Сэнди Метц

Ruby

**Объектно-
ориентированное
проектирование**



Санкт-Петербург · Москва · Екатеринбург · Воронеж
Нижний Новгород · Ростов-на-Дону
Самара · Минск

2017

ББК 32.988.02-018

УДК 004.738.5

М54

Сэнди Метц

М54 Ruby. Объектно-ориентированное проектирование. — СПб.: Питер, 2017. — 304 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-496-02437-2

Мировой бестселлер по программированию на языке Ruby. Книга уже стала классической — с ювелирной точностью она описывает огранку профессионального кода на Ruby. Внимательно изучив это незаменимое руководство, вы сможете:

- Понять, как писать на Ruby качественный код в духе ООП
- Решать, что должно входить в состав класса Ruby
- Не допускать тесной связи между объектами в тех случаях, когда требуется разграничить функциональность
- Определять гибкие интерфейсы между объектами
- Освоить утиную типизацию
- Эффективно задействовать наследование, композицию и полиморфизм
- Разрабатывать экономные тесты
- Доводить до совершенства любой legacy-код Ruby

12+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018

УДК 004.738.5

Права на издание получены по соглашению с Addison-Wesley Longman.

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

ISBN 978-0321721334 англ.

ISBN 978-5-496-02437-2

© Addison-Wesley Professional

© Перевод на русский язык ООО Издательство «Питер», 2017

© Издание на русском языке, оформление ООО Издательство «Питер», 2017

© Серия «Библиотека программиста», 2017

Краткое содержание

Предисловие	13
Введение от научных редакторов	15
Благодарности	20
Об авторе	22
Глава 1. Объектно-ориентированное проектирование	23
Глава 2. Проектирование классов с единственной обязанностью	41
Глава 3. Управление зависимостями	64
Глава 4. Создание гибких интерфейсов	90
Глава 5. Снижение затрат за счет неявной типизации	122
Глава 6. Получение поведения через наследование	145
Глава 7. Разделение ролевого поведения с помощью модулей	184
Глава 8. Объединение объектов путем составления композиции	211
Глава 9. Проектирование экономически эффективных тестов	243
Заключение	300

Оглавление

Предисловие	13
Введение от научных редакторов	15
Кому эта книга будет полезна	16
Как следует читать книгу	17
В чем польза этого издания	18
Благодарности	20
Об авторе	22
Глава 1. Объектно-ориентированное проектирование	23
Хвала проектированию	24
Проблемы, решаемые с помощью проектирования	25
Почему изменения так нелегко даются	26
Определение проектирования	27
Инструменты проектирования	28
Принципы проектирования	28
Шаблоны проектирования	30
Процесс проектирования	30
Когда нужно приступить к проектированию	32
Оценка проектирования	34
Краткое введение в объектно-ориентированное программирование	36
Процедурные языки	36
Объектно-ориентированные языки	37
Выводы	39

Глава 2. Проектирование классов с единственной обязанностью	41
Что должно принадлежать классу.....	42
Группировка методов в классы.....	42
Организация кода для легкого внесения изменений.....	42
Создание классов с единственной обязанностью.....	43
Почему именно единственная обязанность.....	48
Определение наличия у класса единственной обязанности.....	49
Когда следует принимать проектировочные решения.....	50
Создание кода, легко принимающего изменения.....	52
Полагайтесь на поведение, а не на данные.....	52
Повсеместное внедрение единственной обязанности.....	57
И наконец, реальное колесо.....	61
Выводы.....	63
Глава 3. Управление зависимостями	64
Основные сведения о зависимостях.....	65
Выявление зависимостей.....	66
Связи между объектами — Coupling Between Objects (CBO).....	67
Другие зависимости.....	68
Создание кода со слабой связью.....	69
Внедренные зависимости.....	69
Изоляция зависимостей.....	72
Устранение зависимостей от порядка следования аргументов.....	76
Управление направлением зависимостей.....	83
Разворот в обратном направлении.....	83
Выбор направления.....	84
Определение конкретности и абстрактности.....	85
Выводы.....	89
Глава 4. Создание гибких интерфейсов	90
Основные сведения об интерфейсах.....	91
Определение интерфейсов.....	93
Открытые интерфейсы.....	94
Закрытые интерфейсы.....	94
Обязанности, зависимости и интерфейсы.....	94
Поиск открытого интерфейса.....	95
Пример приложения: компания, занимающаяся велотуризмом.....	95
Формирование намерения.....	96

Диаграммы последовательности	97
Нужно не говорить «как», а спрашивать «что»	102
Поиск контекста независимости	105
Доверие, оказываемое другим объектам	108
Сообщения для обнаружения потребности в новых объектах	109
Создание приложения, основанного на сообщениях	112
Написание кода с отличным интерфейсом	112
Создавайте четко выраженные интерфейсы	113
Уважайте чужие открытые интерфейсы	115
Будьте осмотрительны при наличии зависимости от закрытых интерфейсов	116
Минимизация контекста	116
Закон Деметры	117
Определение закона	117
Последствия нарушений	117
Как обойтись без нарушений	119
Прислушиваясь к закону Деметры	120
Выводы	121
Глава 5. Снижение затрат за счет неявной типизации	122
Основные сведения о неявной типизации	123
Упущение из виду возможностей применения неявной типизации	124
Усугубление проблемы	125
Скрытые возможности неявной типизации	128
Последствия неявной типизации	132
Написание кода с использованием неявной типизации	134
Обнаружение скрытых возможностей применения неявной типизации	134
Внедрение доверия в использование неявной типизации	136
Документирование неявных типов	137
Распределение кода между «утками»	137
Мудрый подход к выбору «уток»	138
Преодоление страха применения неявной типизации	139
Подрыв неявной типизации с помощью статической типизации	139
Сравнение статической и динамической типизации	140
Вступление на путь динамической типизации	141
Выводы	144

Глава 6. Получение поведения через наследование	145
Основные сведения о классическом наследовании	146
Как определить, где требуется наследование	147
Начнем с конкретного класса	147
Встраивание нескольких типов	149
Поиск встраиваемых типов	152
Выбор наследования	153
Прорисовка наследственных связей	155
Ошибочное применение наследования	156
Поиск абстракции	158
Создание абстрактного родительского класса	159
Перемещение вверх абстрактного поведения	163
Отделение абстрактного от конкретного	166
Использование схемы шаблонного метода	168
Реализация каждого шаблонного метода	170
Управление связанностью родительских классов и подклассов	172
Общие сведения о связанности	173
Устранение связанности подклассов с использованием хук-сообщений	177
Выводы	182
Глава 7. Разделение ролевого поведения с помощью модулей	184
Основные сведения о ролях	185
Поиск ролей	185
Организация обязанностей	187
Устранение ненужных зависимостей	190
Выявление неявного типа, подходящего для планирования	190
Нужно позволить объектам говорить самим за себя	191
Написание конкретного кода	192
Извлечение абстракции	195
Поиск методов	199
Грубое упрощение	199
Уточненное объяснение	201
Почти полное объяснение	203
Наследование ролевого поведения	205
Написание наследуемого кода	205
Выявление антишаблонов	205

Принуждение к абстракции	206
Соблюдение контракта	207
Использование схемы шаблонного метода	208
Превентивное отделение классов	208
Создание неглубоких иерархий	208
Выводы	210
Глава 8. Объединение объектов путем составления композиции	211
Составление композиции Bicycle (велосипед) из Parts (частей)	212
Обновление класса Bicycle	212
Создание иерархии Parts	214
Составление композиции для объекта Parts	217
Создание Part	217
Придание объекту Parts большей схожести с массивом	221
Изготовление Parts-объектов	225
Создание модуля PartsFactory	226
Применение PartsFactory	228
Bicycle в виде композиции	230
Выбор между наследованием и композицией	234
Приемлемость наследования	234
Приемлемость композиции	237
Выбор характера отношений	239
Выводы	242
Глава 9. Проектирование экономически эффективных тестов	243
Целенаправленное тестирование	244
Осознание намерений	245
Выявление предмета тестирования	247
Умение определять нужный момент для тестирования	251
Умение проводить тестирование	253
Тестирование входящих сообщений	255
Удаление неиспользуемых интерфейсов	257
Проверка открытого интерфейса	258
Изоляция тестируемого объекта	260
Внедрение зависимостей с использованием классов	262
Внедрение зависимостей в качестве ролей	264

Тестирование закрытых методов	270
Игнорирование закрытых методов при тестировании	270
Удаление закрытых методов из тестируемого класса	271
Выбор в пользу тестирования закрытого метода	271
Тестирование исходящих сообщений	273
Игнорирование сообщений-запросов	273
Проверка сообщений-команд	274
Тестирование неявных типов	277
Тестирование ролей	277
Ролевые тесты для проверки дублеров	283
Тестирование унаследованного кода	287
Определение унаследованного интерфейса	287
Определение обязанностей подкласса	291
Тестирование уникального поведения	294
Выводы	299
Заключение	300