

Содержание (сводка)

	Введение	25
1	Добро пожаловать в мир паттернов: знакомство с паттернами	37
2	Объекты в курсе событий: паттерн Наблюдатель	71
3	Украшение объектов: паттерн Декоратор	111
4	Домашняя ОО-выпечка: паттерн Фабрика	141
5	Уникальные объекты: паттерн Одиночка	199
6	Инкапсуляция вызова: паттерн Команда	219
7	Умение приспосабливаться: паттерны Адаптер и Фасад	261
8	Инкапсуляция алгоритмов: паттерн Шаблонный Метод	299
9	Управляемые коллекции: паттерны Итератор и Композитор	337
10	Состояние дел: паттерн Состояние	405
11	Управление доступом к объектам: паттерн Заместитель	449
12	Паттерны паттернов: составные паттерны	517
13	Паттерны в реальном мире: паттерны для лучшей жизни	593
14	Приложение: другие паттерны	627

Содержание (настоящее)

Введение

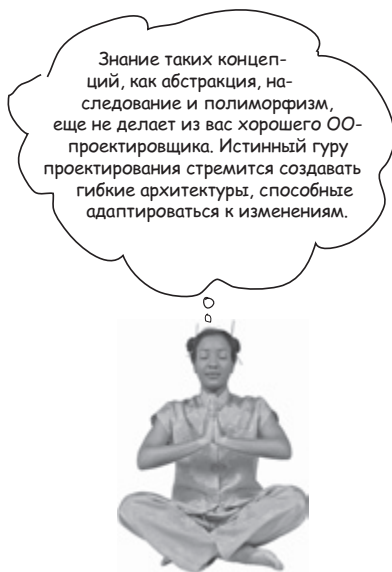
Настройте свой мозг на дизайн паттернов. Вот что вам понадобится, когда вы пытаетесь что-то выучить, в то время как ваш мозг не хочет воспринимать информацию. Ваш мозг считает: «Лучше уж я подумаю о более важных вещах, например об опасных диких животных или почему нельзя голышом прокатиться на сноуборде». Как же заставить свой мозг думать, что ваша жизнь зависит от овладения дизайном паттернов?

Для кого написана эта книга?	26
Мы знаем, о чем вы думаете	27
Метапознание	29
Заставь свой мозг повиноваться	31
Технические рецензенты	34
Благодарности	35

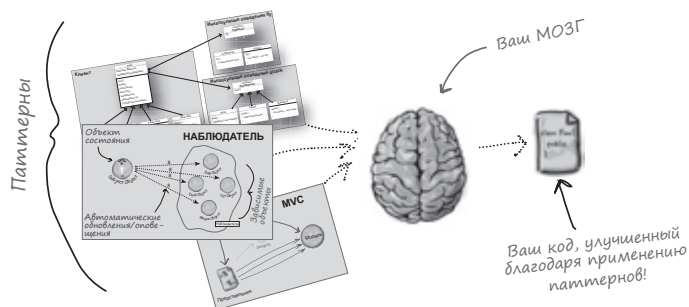
1 Знакомство с паттернами

Добро пожаловать в мир паттернов

Наверняка вашу задачу кто-то уже решил. В этой главе вы узнаете, почему (и как) следует использовать опыт других разработчиков, которые уже сталкивались с аналогичной задачей и успешно решили ее. Заодно мы поговорим об использовании и преимуществах паттернов проектирования, познакомимся с ключевыми принципами ООП и разберем пример одного из паттернов. Лучший способ использовать паттерны — *запомнить их*, а затем научиться *распознавать* те места ваших архитектур и существующих приложений, где их уместно *применить*. Таким образом, вместо программного кода вы повторно используете чужой *опыт*.



Приложение SimUDuck	38
Джо думает о наследовании...	41
Как насчет интерфейса?	42
Единственная константа в программировании	44
Отделяем переменное от постоянного	46
Реализация поведения уток	49
Тестирование кода Duck	54
Динамическое изменение поведения	56
Инкапсуляция поведения: общая картина	58
Отношения СОДЕРЖИТ бывают удобнее отношений ЯВЛЯЕТСЯ	59
Паттерн Стратегия	60
Сила единой номенклатуры	64
Как пользоваться паттернами?	65
Новые инструменты	68
Ответы к упражнениям	69



2 Паттерн Наблюдатель

Объекты в курсе событий

Не упустите, когда происходит что-то интересное! Наш следующий паттерн оповещает объекты о наступлении неких событий, которые могут представлять для них интерес, — причем объекты даже могут решать во время выполнения, желают ли они и дальше получать информацию. Паттерн Наблюдатель чрезвычайно полезен и принадлежит к числу наиболее часто используемых паттернов JDK. Также в этой главе будут рассмотрены связи типа «один-ко-многим» и слабые связи. С паттерном Наблюдатель вы станете душой Общества Паттернов.

Концепции

Абстракция

Реализация

Физм

ание

Принципы

Инкапсулируйте то, что изменяется.

Предпочитайте композицию наследованию.

Программируйте на уровне интерфейсов.

Стремитесь к слабой связанности взаимодействующих объектов.

Приложение Weather Monitoring	73
Знакомство с паттерном Наблюдатель	78
Издатели + Подписчики = Паттерн Наблюдатель	79
Пятиминутная драма: субъект для наблюдения	82
Определение паттерна Наблюдатель	85
Сила слабых связей	87
Проектирование Weather Station	90
Реализация Weather Station	91
Встроенная реализация в языке Java	98
Темная сторона java.util.Observable	105
Новые инструменты	108
Ответы к упражнениям	110

