

Содержание

Посвящение	15
Предисловие	16
Благодарности	17
Об авторе	18
Введение	19
Основная тема книги: код и принципы его разработки	19
Структура книги	21
Как читать эту книгу	22
Рекомендации по обучению с помощью данной книги	25
Часть I. Применение языка	29
Глава 1. Основы использования объектов	31
1.1. Основы: объекты как небольшие и активные сущности	31
1.2. Разработка с помощью объектов	40
1.2.1. Эффективное редактирование кода в среде Eclipse	40
1.2.2. Постепенные конструктивные усовершенствования путем рефакторинга кода	45
1.2.3. Решающее значение именования	48
1.3. Поля	53
1.3.1. Структуры данных	54
1.3.2. Взаимодействующие объекты	56
1.3.3. Свойства	58
1.3.4. Признаки и конфигурация	60
1.3.5. Абстрактное состояние	62
1.3.6. Кеши	63
1.3.7. Обмен данными между методами	66
1.3.8. Статические поля	67
1.4. Методы	69
1.4.1. Объектно-ориентированное представление о методах	70
1.4.2. Перегрузка методов	73
1.4.3. Поставщик услуг	75
1.4.4. Удобные методы	77
1.4.5. Стадии обработки	78
1.4.6. Пояснительные методы	84
1.4.7. События и обратные вызовы	85

1.4.8. Повторно используемые функциональные возможности	87
1.4.9. Шаблонный метод	93
1.4.10. Услуги, делегируемые подклассу	96
1.4.11. Расширение наследуемого поведения	97
1.4.12. Фабричные методы	98
1.4.13. Идентичность, равенство и хеширование	103
1.4.14. Отказ от наследства	106
1.5. Исключения	107
1.5.1. Основное применение исключений	107
1.5.2. Исключения и границы системы	111
1.5.3. Исключения для прояснения программной логики	112
1.5.4. Исключения для проверки предположений	113
1.5.5. Исключения как свидетельство незавершенности	117
1.5.6. Безопасность исключений	118
1.5.7. Проверяемые исключения в сравнении с непроверяемыми	121
1.6. Конструкторы	124
1.6.1. Инициализация объектов	124
1.6.2. Инициализация методами жизненного цикла	127
1.6.3. Конструкторы и наследование	128
1.6.4. Копирование объектов	130
1.6.5. Статические методы-конструкторы	130
1.7. Пакеты	132
1.7.1. Пакеты как компоненты	133
1.7.2. Шаблон “Фасад”	134
1.8. Основы применения классов и объектов	135
1.8.1. Общие характеристики объектов	135
1.8.2. Поставщик услуг	137
1.8.3. Хранители информации	140
1.8.4. Объекты-значения	141
1.8.5. Повторно используемые функциональные возможности	144
1.8.6. Алгоритмы и временные данные	146
1.8.7. Граничные объекты	148
1.8.8. Вложенные классы	149
1.8.9. Нулевой объект	156
Глава 2. Основные структуры объектов	159
2.1. Распространение изменений состояния: шаблон “Наблюдатель”	160
2.1.1. Пример наблюдения за фоновыми заданиями	161
2.1.2. Важные ограничения, накладываемые на проектирование и реализацию	164
2.1.3. Подробности реализации и решения	168
2.1.4. Оценивание потребности в наблюдателях	173

2.2. Составные объекты	175
2.2.1. Владение	176
2.2.2. Совместное пользование структурами	180
2.2.3. Составные объекты и инкапсуляция	181
2.2.4. Составные объекты и наблюдатели	183
2.3. Иерархические структуры	185
2.3.1. Шаблон “Компоновщик”	186
2.3.2. Шаблон “Посетитель”	193
2.3.3. Объекты как языки: шаблон “Интерпретатор”	200
2.3.4. О составных объектах и стековых машинах	207
2.4. Оболочки: адаптеры, заместители и декораторы	212
2.4.1. Шаблон “Адаптер”	213
2.4.2. Шаблон “Декоратор”	216
2.4.3. Шаблон “Заместитель”	217
2.4.4. Оболочки инкапсуляции	219
Глава 3. Абстракция и иерархия	221
3.1. Наследование	221
3.1.1. Принцип подстановки Лисков	222
3.1.2. Интерфейс между базовым классом и подклассами	224
3.1.3. Вынесение общего поведения за скобки	227
3.1.4. Базовые классы для повторно используемой инфраструктуры	228
3.1.5. Базовые классы для абстракции	231
3.1.6. Материализация логики различения вариантов выбора	234
3.1.7. Адаптация поведения	237
3.1.8. Наследование в сравнении с делегированием	238
3.1.9. Операции нисходящего приведения типов и <code>instanceof</code>	240
3.1.10. Наследование реализации	247
3.1.11. Проблема “хрупкого” базового класса	250
3.2. Интерфейсы	256
3.2.1. Поведенческая абстракция	258
3.2.2. Классификация для конкретных клиентов и абстракция	260
3.2.3. Иерархии абстракции	264
3.2.4. Множественная классификация	266
3.2.5. Расширение интерфейсов	267
3.2.6. Указание обратных вызовов	269
3.2.7. Разделение подсистем	270
3.2.8. Интерфейсы-маркеры	275
3.2.9. Управление константами	275
3.2.10. Наследование в сравнении с интерфейсами	276

Часть II. Контракты	279
Глава 4. Контракты объектов	281
4.1. Основы: утверждения плюс инкапсуляция	283
4.2. Выработка принципов на конкретном примере	299
4.2.1. Инварианты и поля модели	301
4.2.2. Контракты в виде полей модели	304
4.2.3. Контракты, инварианты и стадии обработки	306
4.2.4. Роль конструкторов	310
4.2.5. Чистые методы для спецификации	312
4.2.6. Типовые условия: укрощение побочных эффектов	313
4.3. Побудительные причины для составления контрактов	315
4.4. Инварианты и обратные вызовы	317
4.5. Проверка утверждений во время выполнения	320
4.6. Границы системы	322
4.7. Обоснование правильности программ	327
4.7.1. Присваивание	328
4.7.2. Циклы: суммирование элементов массива	334
4.7.3. Условные выражения и циклы: двоичный поиск	340
4.7.4. Краткий обзор некоторых конструкций	345
Глава 5. Тестирование	351
5.1. Основы: модульное тестирование	352
5.2. Принцип предварительного тестирования	359
5.3. Написание и выполнение модульных тестов	364
5.3.1. Основные рекомендации по тестированию	365
5.3.2. Создание тестовых конфигураций	371
5.3.3. Внедрение зависимостей	374
5.3.4. Тестирующие комплекты OSGi	377
5.3.5. Тестирование пользовательского интерфейса	379
5.4. Побуждения к тестированию приложений	383
5.4.1. Тестирование для устранения программных ошибок	383
5.4.2. Тестирование для фиксации контрактов	386
5.4.3. Тестирование для проектирования интерфейса	388
5.4.4. Тестирование для выявления и документирования требований	392
5.4.5. Тестирование для совершенствования проекта	394
5.4.6. Тестирование для документирования достигнутого прогресса	395
5.4.7. Тестирование для обеспечения безопасности	396
5.4.8. Тестирование для внесения изменений	398

5.4.9. Тестирование для уяснения интерфейса прикладного программирования	400
5.4.10. Тестирование для лучшей сбалансированности трудовой деятельности	407
Глава 6. Условия контрактов	411
6.1. Проектирование по контрактам	411
6.1.1. Сначала контракты	412
6.1.2. Менее и более строгие утверждения	413
6.1.3. Толерантные и требовательные стили составления контрактов	415
6.1.4. Практические примеры требовательного стиля составления контрактов	420
6.1.5. Более и менее строгие инварианты классов	422
6.2. Контракты и составные объекты	428
6.2.1. Основы	429
6.2.2. Владение и инварианты	432
6.2.3. Инварианты и совместно используемые объекты	442
6.3. Исключения и контракты	466
6.4. Наследование и создание подтипов	469
6.4.1. Контракты на переопределенные методы	471
6.4.2. Инварианты и наследование	476
Часть III. События	483
Глава 7. Введение в Standard Widget Toolkit	485
7.1. Основы: виджеты, компоновки и события	486
7.2. Графический редактор WindowBuilder для построения графического пользовательского интерфейса	499
7.2.1. Общее представление	500
7.2.2. Создание и запуск SWT-приложений на выполнение	502
7.3. Разработка средствами каркасов	504
7.3.1. Назначение каркасов	505
7.3.2. Инверсия управления	506
7.3.3. Точки адаптации в каркасах	511
7.3.4. Трудности применения каркасов	513
7.4. SWT и платформенно-ориентированный интерфейс	516
7.4.1. Влияние на интерфейс прикладного программирования	516
7.4.2. Влияние, оказываемое на запуск приложений на выполнение	519
7.5. Составные виджеты	520
7.6. Диалоговые окна	531

7.7. Шаблон “Посредник”	537
7.8. Специальное воспроизведение виджетов	541
7.9. Таймеры	546
7.9.1. Задержка и тайм-аут	547
7.9.2. Анимация	551
7.10. Фоновые задания	553
7.10.1. Поток выполнения и пользовательский интерфейс	555
7.10.2. Длительные задания	564
7.10.3. Периодические задания	572
7.11. Краткий обзор событий и контрактов	572
Глава 8. Краткое введение в потоки выполнения	579
8.1. Основы: параллельное выполнение кода	579
8.2. Правильность кода в присутствии потоков выполнения	583
8.3. Обмен уведомлениями между потоками выполнения	592
8.4. Асинхронные сообщения	603
8.5. Открытые вызовы для уведомления	606
8.6. Взаимные блокировки	610
Глава 9. Структурирование приложений с графическим пользовательским интерфейсом	617
9.1. Основы: разделение модели и представления	618
9.2. Шаблон “Модель–представление–контроллер”	630
9.2.1. Основной шаблон	631
9.2.2. Преимущества шаблона “Модель–представление–контроллер”	638
9.2.3. Важные проектные и внедренческие ограничения	641
9.2.4. Типичные недоразумения	644
9.2.5. Поведение на уровне пользовательского интерфейса	645
9.2.6. Наблюдение за моделью со стороны контроллеров	650
9.2.7. Подключаемые контроллеры	651
9.2.8. Вариант “документ–представление” шаблона MVC	654
9.3. Уровень JFace	656
9.3.1. Средства просмотра	657
9.3.2. Завершение шаблона “Модель–представление–контроллер” средствами JFace	670
9.3.3. Привязка данных	672
9.3.4. Меню и действия	681
9.4. Шаблон MVC и прикладной уровень	684
9.4.1. Организация приложения	685

9.4.2. Определение модели	687
9.4.3. Постепенное обновление экрана	690
9.4.4. Логика уровня представления	694
9.5. Отмена и повтор операций	698
9.5.1. Шаблон “Команда”	699
9.5.2. Шаблон “Командный процессор”	707
9.5.3. Затраты труда на отмену и повтор операций	711
9.5.4. Отмена и повтор на практике	713
9.6. Заключительные соображения	718
Глава 10. Конечные автоматы	721
10.1. Основы: состояние и реакция объектов	722
10.2. Конечные автоматы в реальных ситуациях	734
10.2.1. Дополнительные основополагающие элементы	735
10.2.2. Текущая деятельность	743
10.2.3. Вложенные конечные автоматы	747
10.3. Реализация конечных автоматов	751
10.3.1. Практический пример редактора линий	751
10.3.2. Состояния как утверждения	752
10.3.3. Явные состояния	761
10.3.4. Шаблон “Состояние”	764
Часть IV. Проектирование на основе обязанностей	771
Глава 11. Методика проектирования на основе обязанностей	773
11.1. Основы: сети взаимодействующих объектов	774
11.2. Принцип единственной ответственности	797
11.2.1. Основной замысел	797
11.2.2. Принцип единственной ответственности и абстракция	802
11.2.3. Принцип единственной ответственности и изменяемость	804
11.3. Исследование объектов и их обязанностей	807
11.3.1. Механизм для построения графиков функций	807
11.3.2. CRC-карточки	808
11.3.3. Выявление объектов и их обязанностей	810
11.3.4. Подведение итогов	838
11.4. Обязанности и иерархия	839
11.5. Главные цели и основные стратегии	843
11.5.1. Соккрытие информации и инкапсуляция	844
11.5.2. Разделение обязанностей	850
11.5.3. Композиционность	855
11.5.4. Прослеживаемость проектного решения в коде	858

11.5.5. Принцип DRY	860
11.5.6. Принципы SOLID	863
Глава 12. Стратегии проектирования	867
12.1. Связывание и сцепление	868
12.1.1. Связывание и изменение	869
12.1.2. Связывание и общие допущения	875
12.1.3. Сцепление	881
12.1.4. Закон Деметры	884
12.2. Проектирование ради гибкости	888
12.2.1. Методики развязывания	889
12.2.2. Шаблон “Уровни”	912
12.3. Расширяемость	923
12.3.1. Основные методики и соображения	923
12.3.2. Шаблон “Перехватчик”	927
12.3.3. Механизм расширений в среде Eclipse	932
12.3.4. Каналы и фильтры	958
12.4. Повторное использование	964
12.4.1. Трудности повторного использования	965
12.4.2. Пример диалоговых окон мастеров в JFace	967
12.4.3. Построение повторно используемого синтаксического анализатора	973
12.4.4. Стратегии повторного использования	981
Часть V. Приложения	987
Приложение. Работа с модулями, подключаемыми к среде Eclipse	989
A.1. Модульная система для Java на уровне OSGi	991
A.1.1. Структура комплектов OSGi	991
A.1.2. Работа с комплектами в среде Eclipse	995
A.1.3. OSGi в качестве платформы для разработки приложений	1001
A.1.4. Определение целевых платформ	1002
A.2. Запуск подключаемых модулей	1005
A.2.1. Модульные тесты JUnit подключаемых модулей	1005
A.2.2. Дополнения Eclipse	1006
A.2.3. Приложения Eclipse	1008
A.2.4. Установка подключаемых модулей в рабочем пространстве	1008
A.2.5. Программы на Java	1009
A.3. Что делать дальше	1010
Библиография	1013
Предметный указатель	1035