

ДАТА-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Разделяем
объекты
и состояние

Йонатан Шарвит



ДАТА-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Йонатан Шарвит

Санкт-Петербург
«БХВ-Петербург»
2024

УДК 004.4'236
ББК 32.973.26-018.1
Ш25

Шарвит Й.

Ш25 Дата-ориентированное программирование: Пер. с англ. — СПб.: БХВ-Петербург, 2024. — 464 с.: ил.

ISBN 978-5-9775-1924-3

Книга посвящена парадигме DOP (дата-ориентированному программированию), являющейся расширением философии объектно-ориентированного программирования. Предлагается новый взгляд на формирование структур данных и операции над ними в высоконагруженных приложениях. Изложенный материал дает решения сложных проблем, возникающих при управлении состоянием, разделяемыми и распределенными данными, позволяет безопасно организовать параллелизм и конкурентность, а также адаптировать ключевые принципы объектно-ориентированного программирования (полиморфизм, композицию, наследование) к новым задачам, связанным с обработкой больших данных.

*Для аналитиков данных, программистов,
руководителей команд и преподавателей*

УДК 004.4'236
ББК 32.973.26-018.1

Группа подготовки издания:

Руководитель проекта	<i>Олег Сивченко</i>
Зав. редакцией	<i>Людмила Гауль</i>
Перевод с английского	<i>Юлии Стикиной</i>
Редактор	<i>Софья Исакова</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Оформление обложки	<i>Зои Канторович</i>

© 2023 BHV

Authorized translation of the English edition © 2022 Manning Publications.

This translation is published and sold by permission of Manning Publications, the owner of all rights to publish and sell the same.

Авторизованный перевод с английского языка на русский издания © 2022 Manning Publications.

Перевод опубликован и продается с разрешения компании-правообладателя Manning Publications.

Все права защищены.

Подписано в печать 07.12.23.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 37,41.

Тираж 1200 экз. Заказ № 7872

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Отпечатано в АО "Первая Образцовая типография"

филиал "Чеховский Печатный Двор"

142300, Московская область, г. Чехов, ул. Полиграфистов, д. 1

Сайт: www.chpd.ru, E-mail: sales@chpd.ru, тел. 8(499)270-73-59

ISBN 978-1-61729-857-8 (англ.)
ISBN 978-5-9775-1924-3 (рус.)

© Manning Publications, 2022

© Перевод на русский язык, оформление. ООО "БХВ-Петербург",
ООО "БХВ", 2024

Оглавление

Вступительное слово	13
Введение	17
Благодарности.....	18
Об этой книге	19
Кто должен прочитать эту книгу?	19
Как организована эта книга: дорожная карта.....	19
О коде.....	21
Дискуссионный форум <i>liveBook</i>	22
Об авторе.....	23
Об иллюстрации на обложке	23
Персонажи пьесы.....	23
ЧАСТЬ 1. ГИБКОСТЬ	25
Глава 1. Сложность объектно-ориентированного программирования	27
1.1. Дизайн ООП: классический или традиционный?	28
1.1.1. Этап проектирования.....	28
1.1.2. UML 101	30
1.1.3. Объяснение каждой части диаграммы классов.....	33
Класс <i>Library</i>	33
Классы <i>Librarian</i> , <i>Member</i> и <i>User</i>	33
Класс <i>Catalog</i>	35
Класс <i>Book</i>	35
Класс <i>BookItem</i>	35
1.1.4. Этап реализации.....	36
1.2. Источники сложности	36
1.2.1. Множество отношений между классами	38
1.2.2. Непредсказуемое поведение кода.....	40
1.2.3. Нетривиальная сериализация данных	42
1.2.4. Сложные иерархии классов	44
Итоги.....	48

Глава 2. Разделение кода и данных	51
2.1. Две части системы ДОП.....	52
2.2. Объекты данных	54
2.3. Модули кода.....	56
2.4. Системы ДОП просты для понимания.....	61
2.5. Системы ДОП являются гибкими	64
Итоги.....	68
Глава 3. Основные манипуляции с данными	69
3.1. Разработка модели данных	70
3.2. Представление записей в виде карт	74
3.3. Манипулирование данными с помощью универсальных функций.....	81
3.4. Вычисление результатов поиска	85
3.5. Обработка записей различных типов.....	93
Итоги.....	98
Глава 4. Управление состоянием	101
4.1. Несколько версий системных данных	102
4.2. Структурное совместное использование	104
4.3. Реализация структурного совместного использования	110
4.4. Безопасность данных.....	112
4.5. Фиксационный этап изменения	114
4.6. Обеспечение целостности состояния системы.....	116
4.7. Восстановление предыдущих состояний.....	117
Итоги.....	120
Глава 5. Основы контроля конкурентности	123
5.1. Оптимистичный контроль конкурентности.....	124
5.2. Согласование между конкурентными изменениями.....	126
5.3. Сокращение коллекций.....	129
5.4. Структурная разница	131
5.5. Имплементация алгоритма согласования	140
Итоги.....	142
Глава 6. Модульные тесты	145
6.1. Простота дата-ориентированных тестовых кейсов.....	145
6.2. Модульные тесты для кода манипулирования данными.....	147
6.2.1. Дерево вызовов функций	149
6.2.2. Модульные тесты для функций вниз по дереву	150
6.2.3. Модульные тесты для узлов в дереве.....	154
6.3. Модульные тесты для запросов.....	157
6.4. Модульные мутационные тесты	162
Движение вперед	171
Итоги.....	172
ЧАСТЬ 2. МАСШТАБИРУЕМОСТЬ	175
Глава 7. Основы валидации данных	179
7.1. Валидация данных в ДОП.....	179
7.2. Суть JSON-схемы	181

7.3. Гибкость и строгость схемы	188
7.4. Композиция схемы	193
7.5. Сведения о сбоях при валидации данных	197
Итоги.....	202
Глава 8. Расширенный контроль конкурентности	203
8.1. Сложность блокировок.....	203
8.2. Потокбезопасный счетчик с атомами	205
8.3. Потокбезопасный кеш с атомами.....	211
8.4. Управление состоянием с помощью атомов	213
Итоги.....	215
Глава 9. Персистентные структуры данных	217
9.1. Потребность в персистентных структурах данных.....	217
9.2. Эффективность персистентных структур данных	221
9.3. Библиотеки персистентных структур данных	227
9.3.1. Персистентные структуры данных в Java	228
9.3.2. Персистентные структуры данных в JavaScript	229
9.4. Персистентные структуры данных в действии	232
9.4.1. Написание запросов с персистентными структурами данных	232
9.4.2. Операции изменения при работе с персистентными структурами данных	235
9.4.3. Сериализация и десериализация	236
9.4.4. Структурная разница	237
Итоги.....	240
Глава 10. Операции с базой данных	243
10.1. Извлечение данных из базы данных	244
10.2. Хранение данных в базе данных	251
10.3. Простая манипуляция данными.....	254
10.4. Продвинутая обработка данных	258
Итоги.....	266
Глава 11. Веб-сервисы.....	269
11.1. Другой запрос функции.....	270
11.2. Создание внутренностей, подобных внешностям.....	270
11.3. Представление запроса клиента в виде карты.....	273
11.4. Представление ответа сервера в виде карты	276
11.5. Дальнейшая передача информации.....	281
11.6. Расширение результатов поиска в действии	284
11.7. Доставка в срок.....	295
Итоги.....	296
ЧАСТЬ 3. УДОБСТВО СОПРОВОЖДЕНИЯ	297
Глава 12. Расширенная проверка данных	299
12.1. Проверка аргументов функции.....	299
12.2. Проверка возвращаемого значения.....	308
12.3. Расширенная проверка данных.....	310
12.4. Автоматическое создание диаграмм модели данных	314

12.5. Автоматическая генерация модульных тестов на основе схемы.....	316
12.6. Новый подарок.....	324
Итоги.....	326
Глава 13. Полиморфизм.....	327
13.1. Сущность полиморфизма.....	327
13.2. Мультиметоды с единичной отправкой.....	332
13.3. Мультиметоды с множественной отправкой.....	337
13.4. Мультиметоды с динамической отправкой.....	343
13.5. Интеграция мультиметодов: производство.....	346
Итоги.....	351
Глава 14. Расширенная обработка данных.....	353
14.1. Обновление значения на карте с помощью выразительности.....	353
14.2. Манипулирование вложенными данными.....	357
14.3. Использование наилучшего инструмента для работы.....	360
14.4. Легкое разматывание.....	365
Итоги.....	370
Глава 15. Отладка.....	371
15.1. Детерминизм в программировании.....	371
15.2. Репродуцируемость с числами и строками.....	375
15.3. Репродуцируемость с любыми данными.....	379
15.4. Модульные тесты.....	383
15.5. Работа с внешними источниками данных.....	392
Прощание.....	394
Итоги.....	394
Приложение А. Принципы дата-ориентированного программирования.....	397
А.1. Принцип № 1: отделяйте код от данных.....	398
А.1.1. Иллюстрация к принципу № 1.....	399
Нарушение принципа № 1 в ООП.....	399
Нарушение принципа № 1 в ФП.....	400
Соблюдение принципа № 1 в ООП.....	400
Соблюдение принципа № 1 в ФП.....	401
А.1.2. Преимущества принципа № 1.....	401
Преимущество № 1: код может быть повторно использован в различных контекстах.....	402
Преимущество № 2: код может быть протестирован изолированно.....	405
Преимущество № 3: системы, как правило, менее сложны.....	406
А.1.3. Издержки принципа № 1.....	408
Издержка № 1: отсутствие контроля над тем, какой код к каким данным может получить доступ.....	408
Издержка № 2: отсутствие пакетирования.....	408
Издержка № 3: системы состоят из большего количества объектов.....	409
А.1.4. Основная суть принципа № 1.....	409
А.2. Принцип № 2: представляйте данные с помощью обобщенных структур.....	410
А.2.1. Иллюстрация к принципу № 2.....	410
А.2.2. Преимущества принципа № 2.....	411

Применение функций, которые не ограничены определенным случаем использования.....	412
Гибкая модель данных.....	412
A.2.3. Издержки принципа № 2	413
Издержка № 1: снижение производительности	413
Издержка № 2: отсутствие схемы данных.....	414
Издержка № 3: отсутствует проверка данных на валидность во время компиляции	414
Издержка № 4: необходимость явного приведения типов.....	415
A.2.4. Основная суть принципа № 2.....	416
A.3. Принцип № 3: данные неизменяемы	417
A.3.1. Иллюстрация к принципу № 3	417
A.3.2. Преимущества принципа № 3	419
Преимущество № 1: надежный доступ к данным для всех.....	419
Преимущество № 2: предсказуемое поведение кода.....	419
Преимущество № 3: быстрая проверка равенства.....	420
Преимущество № 4: безопасность конкурентности без затрат	420
A.3.3. Издержки принципа № 3	420
Издержка № 1: снижается производительность	421
Издержка № 2: требуется библиотека для персистентных структур данных.....	421
A.3.4. Основная суть принципа № 3.....	421
A.4. Принцип № 4: отделяйте схему данных от представления данных.....	422
A.4.1. Иллюстрация к принципу № 4	422
A.4.2. Преимущества принципа № 4	424
Преимущество № 1: возможность свободно выбирать, какие данные следует валидировать.....	424
Преимущество № 2: наличие опциональных полей	426
Преимущество № 3: наличие расширенных условий валидации данных.....	428
Преимущество № 4: возможность автоматического создания визуализации модели данных.....	428
A.4.3. Издержки принципа № 4	429
Издержка № 1: слабая связь между данными и их схемой	430
Издержка № 2: небольшое снижение производительности.....	430
A.4.4. Основная суть принципа № 4.....	430
Заключение.....	431

Приложение В. Обобщенный доступ к данным

в статически типизированных языках	433
V.1. Динамические геттеры для строковых карт	433
V.1.1. Доступ к невложенным полям карты с помощью динамических геттеров.....	434
V.1.2. Доступ к вложенным полям карты с помощью динамических геттеров.....	435
V.2. Геттеры значений для карт.....	436
V.2:1. Доступ к невложенным полям карты с помощью геттеров значений	437
V.2.2. Доступ к вложенным полям карты с помощью геттеров значений	438
V.3. Типизированные геттеры для карт	440
V.3.1. Доступ к невложенным полям карты с помощью типизированных геттеров.....	440
V.3.2. Доступ к вложенным полям карты с помощью типизированных геттеров	441

В.4. Обобщенный доступ к членам класса	443
В.4.1. Обобщенный доступ к не вложенным членам класса	443
В.4.2. Обобщенный доступ к членам вложенного класса	447
В.4.3. Автоматическая сериализация объектов JSON	450
Итого.....	451

Приложение С. Дата-ориентированное программирование:

звено в цепи парадигм программирования	453
С.1. Хронология.....	453
С.1.1. 1958 год: Lisp.....	453
С.1.2. 1981 год: значения и объекты	453
С.1.3. 2000 год: идеальные хеш-деревья.....	455
С.1.4. 2006 год: «Из ямы со смолой»	455
С.1.5. 2007 год: Clojure	455
С.1.6. 2009 год: неизменяемость для всех	455
С.2. Принципы ДОП как наилучший подход	456
С.2.1. Принцип № 1: отделяйте код от данных	456
С.2.2. Принцип № 2: представляйте данные с помощью обобщенных структур.....	456
С.2.3. Принцип № 3: данные неизменяемы	456
С.2.4. Принцип № 4: отделяйте схему данных от представления данных	457
С.3. ДОП и другие парадигмы, связанные с данными	458
С.3.1. Дата-ориентированная разработка.....	458
С.3.2. Дата-управляемое программирование.....	458
С.3.3. Дата-ориентированное программирование (ДОП).....	459
Итого.....	459

Приложение D. Ссылки на Lodash..... 461