

Оглавление

Предисловие	13
Введение.....	15
Благодарности.....	17
О книге.....	18
Об авторе.....	22
Об обложке.....	23
От издательства.....	24

Часть I. Отправная точка

Глава 1. Свойства кода и постановка задачи.....	26
1.1. Свойства кода	26
1.1.1. Внутренние и внешние свойства	27
1.1.2. Функциональные и нефункциональные свойства	28
1.2. Преимущественно внешние свойства	29
1.2.1. Правильность	30
1.2.2. Стабильность.....	30
1.2.3. Удобство использования.....	31
1.2.4. Эффективность.....	31
1.3. Преимущественно внутренние свойства кода	32
1.3.1. Удобочитаемость	32
1.3.2. Возможность повторного использования	32

6 Оглавление

1.3.3. Удобство тестирования	33
1.3.4. Удобство сопровождения	33
1.4. Взаимодействие свойств кода.....	34
1.5. Специфичные свойства.....	36
1.5.1. Потокбезопасность.....	36
1.5.2. Лаконичность	37
1.6. Сквозной пример: система резервуаров для воды.....	37
1.6.1. API.....	38
1.6.2. Сценарий использования.....	39
1.7. Модель данных и представления.....	41
1.7.1. Хранение объема воды	42
1.7.2. Хранение информации о соединениях.....	43
1.8. Hello, резервуары! [Novice].....	45
1.8.1. Поля и конструктор.....	45
1.8.2. Методы getAmount и addWater	47
1.8.3. Метод connectTo	47
Итоги.....	50
Дополнительная литература.....	50
Глава 2. Эталонная реализация	51
2.1. Код [Reference].....	53
2.1.1. Диаграммы распределения памяти.....	54
2.1.2. Методы.....	57
2.2. Требования к памяти.....	59
2.2.1. Требования к памяти в Reference.....	61
2.3. Временная сложность.....	63
2.3.1. Временная сложность в Reference.....	67
2.4. Применим изученное.....	67
Итоги.....	68
Ответы на вопросы и упражнения.....	69
Дополнительная литература.....	72
Часть II. Свойства программного кода	
Глава 3. Жажда скорости: эффективность по времени	74
3.1. Добавление воды с постоянным временем [Speed1]	76
3.1.1. Временная сложность.....	78

3.2. Добавление соединений за постоянное время [Speed2]	79
3.2.1. Представление групп в виде циклических списков	79
3.2.2. Отложенные обновления	82
3.3. Лучший баланс: алгоритмы поиска объединений [Speed3]	85
3.3.1. Поиск представителя группы	87
3.3.2. Соединение деревьев	90
3.3.3. Временная сложность в наихудшем случае	91
3.3.4. Амортизированная временная сложность	93
3.3.5. Амортизированный анализ массивов с изменяемым размером	95
3.4. Сравнение реализаций	98
3.4.1. Эксперименты	99
3.4.2. Теория и практика	101
3.5. А теперь совсем другое	102
3.5.1. Быстрая вставка	103
3.5.2. Быстрые запросы	104
3.5.3. Все операции	104
3.6. Реальные сценарии использования	105
3.7. Применим изученное	106
Итоги	107
Ответы на вопросы и упражнения	108
Дополнительная литература	111
Глава 4. Эффективность по затратам памяти	113
4.1. Первые шаги [Memory1]	114
4.1.1. Временная сложность и затраты памяти	117
4.2. Простые массивы	119
4.2.1. Временная сложность и затраты памяти	121
4.3. Отказ от объектов [Memory3]	124
4.3.1. API без объектов	124
4.3.2. Поля и метод getAmount	127
4.3.3. Создание резервуаров фабричным методом	128
4.3.4. Соединение резервуаров по идентификаторам	131
4.3.5. Временная сложность и затраты памяти	135
4.4. Черная дыра [Memory4]	135
4.4.1. Временная сложность и затраты памяти	139
4.5. Баланс затрат памяти и времени	139

8 Оглавление

4.6. А теперь совсем другое.....	141
4.6.1. Малое количество дубликатов	142
4.6.2. Большое количество дубликатов.....	143
4.7. Реальные сценарии использования	144
4.8. Применим изученное.....	145
Итоги.....	147
Ответы на вопросы и упражнения.....	147
Дополнительная литература.....	152
Глава 5. Надежность за счет мониторинга.....	154
5.1. Контрактное проектирование.....	154
5.1.1. Предусловия и постусловия.....	155
5.1.2. Инварианты	158
5.1.3. Правильность и стабильность	158
5.1.4. Проверка контрактов.....	160
5.1.5. Более широкая картина	162
5.2. Контрактное проектирование резервуаров.....	164
5.3. Резервуары, проверяющие свои контракты [Contracts]	167
5.3.1. Проверка контракта addWater.....	167
5.3.2. Проверка контракта connectTo.....	171
5.4. Резервуары, проверяющие свои инварианты [Invariants].....	174
5.4.1. Проверка инвариантов в connectTo	176
5.4.2. Проверка инвариантов в addWater	178
5.5. А теперь совсем другое.....	179
5.5.1. Контракты.....	180
5.5.2. Эталонная реализация	180
5.5.3. Проверка контрактов.....	181
5.5.4. Проверка инвариантов.....	182
5.6. Реальные сценарии использования	184
5.7. Применим изученное.....	185
Итоги.....	187
Ответы на вопросы и упражнения.....	187
Дополнительная литература.....	193
Глава 6. Надежность за счет тестирования.....	194
6.1. Основные понятия тестирования.....	195
6.1.1. Покрытие в тестировании.....	195

6.1.2. Тестирование и контрактное проектирование.....	196
6.1.3. JUnit.....	198
6.2. Тестирование резервуаров [UnitTests]	200
6.2.1. Инициализация тестов.....	200
6.2.2. Тестирование addWater.....	202
6.2.3. Тестирование connectTo.....	209
6.2.4. Выполнение тестов.....	211
6.2.5. Оценка покрытия кода.....	212
6.3. Тестируемость [Testable].....	214
6.3.1. Управляемость	214
6.3.2. Наблюдаемость	215
6.3.3. Isolation: разделение зависимостей.....	216
6.4. А теперь совсем другое.....	218
6.4.1. Повышение удобства тестирования	219
6.4.2. Набор тестов	221
6.5. Реальные сценарии использования	224
6.6. Применим изученное.....	225
Итоги.....	226
Ответы на вопросы и упражнения.....	226
Дополнительная литература.....	233
Глава 7. Удобочитаемость	234
7.1. Разный взгляд на удобочитаемость	234
7.1.1. Корпоративные руководства по стилю программирования	236
7.1.2. Составляющие удобочитаемости	237
7.2. Структурные факторы удобочитаемости	238
7.2.1. Управляющие команды.....	239
7.2.2. Выражения и локальные переменные	241
7.3. Внешние факторы удобочитаемости.....	242
7.3.1. Комментарии	243
7.3.2. Выбор имен	244
7.3.3. Пропуски и отступы.....	245
7.4. Код [Readable].....	246
7.4.1. Документирование заголовка класса комментариями Javadoc.....	247
7.4.2. Чистка метода connectTo	251
7.4.3. Чистка addWater.....	255

- 7.5. Напоследок об удобочитаемости..... 257
- 7.6. А теперь совсем другое..... 258
- 7.7. Реальные сценарии использования 260
- 7.8. Применим изученное..... 261
- Итоги..... 263
- Ответы на вопросы и упражнения..... 263
- Дополнительная литература..... 267
- Глава 8. Потокобезопасность..... 269**
 - 8.1. Проблемы потокобезопасности 270
 - 8.1.1. Уровни конкурентности..... 272
 - 8.1.2. Политика конкурентности для резервуаров 275
 - 8.2. Взаимоблокировки 276
 - 8.2.1. Атомарные последовательности блокировок..... 277
 - 8.2.2. Упорядоченные последовательности блокировок..... 279
 - 8.2.3. Скрытое состояние гонки..... 281
 - 8.3. Потокобезопасные резервуары [ThreadSafe] 282
 - 8.3.1. Синхронизация connectTo 283
 - 8.3.2. Синхронизация addWater и getAmount 284
 - 8.4. Неизменяемость 287
 - 8.4.1. API..... 289
 - 8.4.2. Реализация 290
 - 8.5. А теперь совсем другое..... 294
 - 8.6. Реальные сценарии использования 296
 - 8.7. Применим изученное..... 297
 - Итоги..... 299
 - Ответы на вопросы и упражнения..... 299
 - Дополнительная литература..... 302
- Глава 9. Повторное использование 304**
 - 9.1. Определение границ..... 304
 - 9.2. Общая структура..... 306
 - 9.2.1. API атрибутов..... 310
 - 9.2.2. Изменяемые коллекторы..... 311
 - 9.2.3. Адаптация Attribute для функциональных интерфейсов..... 317
 - 9.3. Обобщенная реализация резервуара..... 318
 - 9.4. Общие соображения..... 320

9.5. Использование готового кода [Generic].....	321
9.5.1. Обновленный сценарий использования	321
9.5.2. Проектирование конкретного атрибута.....	322
9.5.3. Определение класса для резервуара.....	324
9.6. Посты в соцсетях.....	324
9.7. А теперь совсем другое.....	325
9.7.1. Интерфейс для параметрических функций.....	327
9.7.2. Схема коммуникаций	329
9.8. Реальные сценарии использования	332
9.9. Применим изученное.....	334
Итоги.....	336
Ответы на вопросы и упражнения.....	336
Дополнительная литература.....	342

Приложения

Приложение А. Программный гольф: компактность.....	344
А.1. Самая короткая программа, которая у меня получилась [Golfing]	344
Дополнительная литература.....	347
Приложение Б. Финальная версия класса для резервуара с водой	348
Б.1. Улучшения удобочитаемости.....	349
Б.2. Улучшения надежности.....	350