

Содержание

Предисловие	10
Введение	11
Безопасное программирование на Java	12
Предмет книги	13
Благодарности	17
Об авторах	19
От издательства	22
Глава 1. Безопасность	23
1. Ограничивайте срок действия уязвимых данных	24
2. Не храните уязвимые данные незашифрованными на стороне клиента	27
3. Снабжайте уязвимые изменяемые классы немодифицируемыми оболочками	30
4. Вызывайте уязвимые для безопасности методы с проверенными аргументами	32
5. Не допускайте выгрузку произвольных файлов	33
6. Кодировать или экранировать выводимые данные надлежащим образом	36
7. Предотвращайте внедрение кода	40
8. Предотвращайте внедрение операторов XPath	42
9. Предотвращайте внедрение операторов LDAP	46
10. Не пользуйтесь методом <code>clone()</code> для копирования небезопасных параметров метода	49
11. Не пользуйтесь методом <code>Object.equals()</code> для сравнения ключей шифрования	52
12. Не пользуйтесь небезопасными или слабыми алгоритмами шифрования	53
13. Храните пароли с помощью хеш-функции	54
14. Обеспечьте подходящее начальное случайное значение для класса <code>SecureRandom</code>	59
15. Не полагайтесь на методы, которые могут быть переопределены в ненадежном коде	60
16. Старайтесь не предоставлять излишние полномочия	66
17. Сводите к минимуму объем привилегированного кода	69
18. Не раскрывайте методы с нестрогими проверками ненадежного кода	70
19. Определяйте специальные полномочия доступа для мелкоструктурной защиты	78
20. Создавайте безопасную “песочницу”, используя диспетчер защиты	81
21. Не допускайте злоупотреблений привилегиями методов обратного вызова в ненадежном коде	85
Глава 2. Защитное программирование	91
22. Минимизируйте область действия переменных	92
24. Минимизируйте доступность классов и их членов	96

25. Документируйте потоковую безопасность и пользуйтесь аннотациями везде, где только можно	100
26. Всегда предоставляйте отклик на результирующее значение метода	106
27. Распознавайте файлы, используя несколько файловых атрибутов	109
28. Не присоединяйте значимость к порядковому значению, связанному с перечислением	115
29. Принимайте во внимание числовое продвижение типов	117
30. Активизируйте проверку типов в методах с переменным количеством аргументов во время компиляции	121
31. Не объявляйте открытыми и конечными константы, значения которых могут измениться в последующих выпусках программы	123
32. Избегайте циклических зависимостей пакетов	126
33. Отдавайте предпочтение определяемым пользователем исключениям над более общими типами исключений	128
34. Старайтесь изящно исправлять системные ошибки	130
35. Тщательно разрабатывайте интерфейсы, прежде чем их выпускать	132
36. Пишите код, удобный для “сборки мусора”	135
Глава 3. Надежность	139
37. Не затеняйте и не заслоняйте идентификаторы в подобластях действия	140
38. Не указывайте в одном объявлении больше одной переменной	142
39. Пользуйтесь описательными символическими константами для обозначения литеральных значений в логике программы	145
40. Правильно кодируйте отношения в определениях констант	148
41. Возвращайте из методов пустой массив или коллекцию вместо пустого значения	149
42. Пользуйтесь исключениями только в особых случаях	152
43. Пользуйтесь оператором <code>try</code> с ресурсами для безопасного обращения с закрываемыми ресурсами	154
44. Не пользуйтесь утверждениями для проверки отсутствия ошибок при выполнении	157
45. Пользуйтесь вторым и третьим однотипными операндами в условных выражениях	158
46. Не выполняйте сериализацию прямых описателей системных ресурсов	162
47. Отдавайте предпочтение итераторам над перечислениями	164
48. Не пользуйтесь прямыми буферами для хранения нечасто используемых объектов с коротким сроком действия	166
49. Удаляйте объекты с коротким сроком действия из контейнерных объектов с длительным сроком действия	167
Глава 4. Понятность программ	171
50. Будьте внимательны, применяя визуально дезориентирующие идентификаторы и литералы	171
51. Избегайте неоднозначной перегрузки методов с переменным количеством аргументов	175
52. Избегайте внутренних индикаторов ошибок	177
53. Не выполняйте операции присваивания в условных выражениях	179

54. Пользуйтесь фигурными скобками в теле условного оператора <code>if</code> , а также циклов <code>for</code> или <code>while</code>	181
55. Не ставьте точку с запятой сразу после условного выражения с оператором <code>if</code> , <code>for</code> или <code>while</code>	183
56. Завершайте каждый набор операторов, связанных с меткой <code>case</code> , оператором <code>break</code>	184
57. Избегайте неумышленного зацикливания счетчиков циклов	186
58. Пользуйтесь круглыми скобками для обозначения операций предшествования	188
59. Не делайте никаких предположений о создании файлов	190
60. Преобразуйте целые значения в значения с плавающей точкой для выполнения операций с плавающей точкой	192
61. Непременно вызывайте метод <code>super.clone()</code> из метода <code>clone()</code>	195
62. Употребляйте комментарии единообразно и в удобном для чтения виде	197
63. Выявляйте и удаляйте излишний код и значения	198
64. Стремитесь к логической полноте	202
65. Избегайте неоднозначной или вносящей путаницу перегрузки	205
Глава 5. Ложные представления программистов	209
66. Не принимайте на веру, что объявление изменчивой ссылки гарантировало надежную публикацию членов объекта, доступного по этой ссылке	209
67. Не принимайте на веру, что методы <code>sleep()</code> , <code>yield()</code> или <code>getState()</code> предоставляли семантику синхронизации	215
68. Не принимайте на веру, что оператор вычисления остатка всегда возвращал неотрицательный результат для целочисленных операндов	219
69. Не путайте равенство абстрактных объектов с равенством ссылок	220
70. Ясно различайте поразрядные и логические операторы	223
71. Правильно интерпретируйте управляющие символы при загрузке строк	226
72. Не пользуйтесь перегружаемыми методами для динамического различения типов данных	229
73. Не путайте неизменяемость ссылки и доступного по ссылке объекта	231
74. Аккуратно пользуйтесь методами сериализации <code>writeUnshared()</code> и <code>readUnshared()</code>	235
75. Не пытайтесь оказывать помощь системе “сборки мусора”, устанавливая пустое значение в локальных переменных ссылочного типа	239
Приложение А. Android	241
Приложение Б. Словарь специальных терминов	245
Приложение В. Библиография	249
Предметный указатель	255