

Spis treści

Słowo wstępne (9)

Przedmowa (11)

Podziękowania (15)

Rozdział 1. Wprowadzenie (19)

Rozdział 2. Tworzenie i usuwanie obiektów (23)

Temat 1. Tworzenie statycznych metod fabrycznych zamiast konstruktorów (23)

Temat 2. Zastosowanie budowniczego do obsługi wielu parametrów konstruktora (28)

Temat 3. Wymuszanie właściwości singleton za pomocą prywatnego konstruktora lub typu enum (36)

Temat 4. Wykorzystanie konstruktora prywatnego w celu uniemożliwienia utworzenia obiektu (38)

Temat 5. Stosuj wstrzykiwanie zależności zamiast odwoływania się do zasobów na sztywno (39)

Temat 6. Unikanie powielania obiektów (41)

Temat 7. Usuwanie niepotrzebnych referencji do obiektów (45)

Temat 8. Unikanie finalizatorów i oczyszczaczy (48)

Temat 9. Preferuj konstrukcję try z zasobami zamiast try-finally (54)

Rozdział 3. Metody wspólne dla wszystkich obiektów (57)

Temat 10. Zachowanie założeń w trakcie przeddefiniowywania metody equals (58)

Temat 11. Przeddefiniowywanie metody hashCode wraz z equals (70)

Temat 12. Przeddefiniowywanie metody toString (75)

Temat 13. Rozsądne przeddefiniowywanie metody clone (78)

Temat 14. Implementacja interfejsu Comparable (86)

Rozdział 4. Klasy i interfejsy (93)

Temat 15. Ograniczanie dostępności klas i ich składników (93)

Temat 16. Stosowanie metod akcesorów zamiast pól publicznych w klasach publicznych (98)

Temat 17. Zapewnianie niezmienności obiektu (100)

Temat 18. Zastępowanie dziedziczenia kompozycją (107)

Temat 19. Projektowanie i dokumentowanie klas przeznaczonych do dziedziczenia (113)

Temat 20. Stosowanie interfejsów zamiast klas abstrakcyjnych (119)

Temat 21. Projektowanie interfejsów na długie lata (124)

Temat 22. Wykorzystanie interfejsów jedynie do definiowania typów (127)

Temat 23. Zastępowanie oznaczania klas hierarchią (129)

Temat 24. Zalety stosowania statycznych klas składowych (132)

Temat 25. Ograniczenie pliku źródłowego do pojedynczej klasy głównego poziomu (135)

Rozdział 5. Typy ogólne (139)

Temat 26. Nie korzystaj z typów surowych (139)

Temat 27. Eliminowanie ostrzeżeń o braku kontroli (144)

Temat 28. Korzystanie z list zamiast tablic (147)

Temat 29. Stosowanie typów ogólnych (151)

Temat 30. Stosowanie metod ogólnych (156)

Temat 31. Zastosowanie związanych szablonów do zwiększania elastyczności API (159)

Temat 32. Ostrożne łączenie typów ogólnych i parametrów varargs (166)

Temat 33. Wykorzystanie heterogenicznych kontenerów bezpiecznych dla typów (171)

Rozdział 6. Typy wyliczeniowe i adnotacje (177)

Temat 34. Użycie typów wyliczeniowych zamiast stałych int (177)

- Temat 35. Użycie pól instancyjnych zamiast kolejności (188)
- Temat 36. Użycie EnumSet zamiast pól bitowych (189)
- Temat 37. Użycie EnumMap zamiast indeksowania kolejnością (191)
- Temat 38. Emulowanie rozszerzalnych typów wyliczeniowych za pomocą interfejsów (196)
- Temat 39. Korzystanie z adnotacji zamiast wzorców nazw (200)
- Temat 40. Spójne użycie adnotacji Override (207)
- Temat 41. Użycie interfejsów znacznikowych do definiowania typów (210)

Rozdział 7. Lambdy i strumienie (213)

- Temat 42. Stosuj lambdy zamiast klas anonimowych (213)
- Temat 43. Wybieraj referencje do metod zamiast lambd (217)
- Temat 44. Korzystaj ze standardowych interfejsów funkcyjnych (219)
- Temat 45. Rozważnie korzystaj ze strumieni (223)
- Temat 46. Stosuj w strumieniach funkcje bez efektów ubocznych (231)
- Temat 47. Zwracaj kolekcje zamiast strumieni (236)
- Temat 48. Ostrożnie korzystaj ze strumieni równoległych (242)

Rozdział 8. Metody (247)

- Temat 49. Sprawdzanie poprawności parametrów (247)
- Temat 50. Defensywne kopiowanie (250)
- Temat 51. Projektowanie sygnatur metod (255)
- Temat 52. Rozsądne korzystanie z przeciążania (257)
- Temat 53. Rozsądne korzystanie z metod varargs (263)
- Temat 54. Zwracanie pustych tablic lub kolekcji zamiast wartości null (265)
- Temat 55. Rozsądne zwracanie obiektów opcjonalnych (267)
- Temat 56. Tworzenie komentarzy dokumentujących dla udostępnianych elementów API (272)

Rozdział 9. Programowanie (281)

- Temat 57. Ograniczanie zasięgu zmiennych lokalnych (281)
- Temat 58. Stosowanie pętli for-each zamiast tradycyjnych pętli for (284)
- Temat 59. Poznanie i wykorzystywanie bibliotek (287)
- Temat 60. Unikanie typów float i double, gdy potrzebne są dokładne wyniki (290)
- Temat 61. Stosowanie typów prostych zamiast opakowanych typów prostych (292)
- Temat 62. Unikanie typu String, gdy istnieją bardziej odpowiednie typy (296)
- Temat 63. Problemy z wydajnością przy łączeniu ciągów znaków (298)
- Temat 64. Odwoływanie się do obiektów poprzez interfejsy (299)
- Temat 65. Stosowanie interfejsów zamiast refleksyjności (301)
- Temat 66. Rozważne wykorzystywanie metod natywnych (304)
- Temat 67. Unikanie przesadnej optymalizacji (306)
- Temat 68. Wykorzystanie ogólnie przyjętych konwencji nazewnictwa (309)

Rozdział 10. Wyjątki (313)

- Temat 69. Wykorzystanie wyjątków w sytuacjach nadzwyczajnych (313)
- Temat 70. Stosowanie wyjątków przechwytywanych i wyjątków czasu wykonania (316)
- Temat 71. Unikanie niepotrzebnych wyjątków przechwytywanych (318)
- Temat 72. Wykorzystanie wyjątków standardowych (320)
- Temat 73. Zgłaszanie wyjątków właściwych dla abstrakcji (323)
- Temat 74. Dokumentowanie wyjątków zgłaszanych przez metodę (325)
- Temat 75. Udostępnianie danych o błędzie (326)
- Temat 76. Zachowanie atomowości w przypadku błędu (328)
- Temat 77. Nie ignoruj wyjątków (330)

Rozdział 11. Współbieżność (333)

- Temat 78. Synchronizacja dostępu do wspólnych modyfikowalnych danych (333)

Temat 79. Unikanie nadmierowej synchronizacji (338)

Temat 80. Stosowanie wykonawców, zadań i strumieni zamiast wątków (344)

Temat 81. Stosowanie narzędzi współbieżności zamiast wait i notify (346)

Temat 82. Dokumentowanie bezpieczeństwa dla wątków (352)

Temat 83. Rozsądne korzystanie z późnej inicjalizacji (355)

Temat 84. Nie polegaj na harmonogramie wątków (358)

Rozdział 12. Serializacja (361)

Temat 85. Stosuj rozwiązania alternatywne wobec serializacji Javy (361)

Temat 86. Rozsądne implementowanie interfejsu Serializable (365)

Temat 87. Wykorzystanie własnej postaci serializowanej (368)

Temat 88. Defensywne tworzenie metody readObject (375)

Temat 89. Stosowanie typów wyliczeniowych zamiast readResolve do kontroli obiektów (381)

Temat 90. Użycie pośrednika serializacji zamiast serializowanych obiektów (385)

Dodatek A. Tematy odpowiadające drugiemu wydaniu (389)

Dodatek B. Zasoby (393)

Skorowidz (399)