

Spis treści |

Przedmowa	11
O autorze	13
O korektorach merytorycznych	13
Wprowadzenie	17
ROZDZIAŁ 1	
Łatwa obsługa techniczna	21
Co w ogóle oznacza łatwa obsługa techniczna?	21
Łatwa obsługa techniczna pozwala na większą funkcjonalność	23
Łatwa obsługa techniczna to zadowolenie programisty	25
Łatwość obsługi technicznej ułatwia podejmowanie decyzji	26
Zachowanie łatwości obsługi technicznej	27
ROZDZIAŁ 2	
Na czym polega problem z warstwami?	30
Warstwy wspierają projekt oparty na bazie danych	31
Warstwy są podatne na skróty	33
Warstwy utrudniają testowanie	34
Warstwy ukrywają przypadki użycia	35
Warstwy utrudniają pracę równoległą	36
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	37
ROZDZIAŁ 3	
Odwracanie zależności	38
Reguła jednej odpowiedzialności	38
Opowieść o efektach ubocznych	39
Zasada odwrócenia zależności	40
Czysta architektura	41

Architektura heksagonalna	43
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	45

ROZDZIAŁ 4

Organizowanie kodu	47
Organizacja kodu za pomocą warstw	47
Organizacja kodu za pomocą funkcjonalności	48
Architekturalnie ekspresyjna struktura pakietu	49
Rola wstrzykiwania zależności	52
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	53

ROZDZIAŁ 5

Implementowanie przypadku użycia	54
Implementowanie modelu dziedziny	54
Krótki opis przypadku użycia	56
Weryfikowanie danych wejściowych	58
Potężne konstruktory	60
Różne modele danych wejściowych dla różnych przypadków użycia	62
Weryfikowanie reguł biznesowych	63
Rozbudowany kontra uproszczony model dziedziny	65
Różne modele danych wyjściowych dla różnych przypadków użycia	66
Przypadki użycia przeznaczone tylko do odczytu	67
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	68

ROZDZIAŁ 6

Implementowanie adaptera internetowego	69
Odwrócenie zależności	69
Zadania adaptera internetowego	71
Kontrolery wycinków adaptera internetowego	72
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	76

ROZDZIAŁ 7

Implementowanie adaptera trwałego magazynu danych	77
Odwrócenie zależności	77
Zadania adaptera trwałego magazynu danych	78
Dzielenie interfejsów portu	79

Dzielenie adapterów trwałego magazynu danych	81
Przykład oparty na JPA Spring Data	82
Transakcje bazy danych	87
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	88

ROZDZIAŁ 8

Testowanie elementów architektury	89
Piramida testów	89
Testowanie encji dziedziny za pomocą testów jednostkowych	91
Testowanie przypadku użycia za pomocą testu jednostkowego	92
Testowanie adaptera internetowego za pomocą testów integracyjnych	94
Testowanie adaptera trwałego magazynu danych za pomocą testów integracyjnych	95
Testowanie ścieżek głównych za pomocą testów systemowych	97
Jaka liczba testów będzie wystarczająca?	101
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	103

ROZDZIAŁ 9

Mapowanie między granicami	104
Strategia braku mapowania	104
Strategia mapowania dwukierunkowego	106
Strategia mapowania pełnego	107
Strategia mapowania jednokierunkowego	109
Kiedy należy używać poszczególnych rodzajów mapowania?	110
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	111

ROZDZIAŁ 10

Złożenie aplikacji w całość	112
Dlaczego złożenie wszystkiego w całość ma znaczenie?	112
Połączenie elementów za pomocą zwykłego kodu	114
Złożenie aplikacji poprzez skanowanie ścieżki classpath przeprowadzane przez Springa	115
Złożenie aplikacji poprzez konfigurację Javy w Springu	117
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	119

ROZDZIAŁ 11

Rozsądne używanie skrótów	121
Dlaczego skrót przypomina wybitą szybę?	121
Odpowiedzialność za dobry początek	122
Współdzielenie modeli między przypadkami użycia	123
Używanie encji dziedziny jako modeli danych wejściowych lub wyjściowych	124
Pomijanie portów wejściowych	125
Pomijanie usług	126
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	127

ROZDZIAŁ 12

Egzekwowanie granic architektury	129
Granice i zależności	129
Modyfikatory widoczności	130
Funkcja przystosowania wykonywana po przeprowadzeniu kompilacji	132
Artefakty kompilacji	134
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	138

ROZDZIAŁ 13

Zarządzanie wieloma ograniczonymi kontekstami	139
Jeden sześciokąt dla ograniczonego kontekstu?	140
Rozdzielone ograniczone konteksty	142
Poprawne połączenie ograniczonych kontekstów	144
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	146

ROZDZIAŁ 14**Podejście do architektury oprogramowania**

oparte na komponentach	147
Modułowość dzięki komponentom	148
Przykład: tworzenie komponentu typu „silnik sprawdzania”	150
Egzekwowanie granic komponentów	153
W jaki sposób może to pomóc w tworzeniu oprogramowania łatwego w późniejszej obsłudze technicznej?	154

ROZDZIAŁ 15

Wybór stylu architektonicznego	155
Rozpocznij od prostego rozwiązania	155
Ewolucja dziedziny	156
Zaufaj swojemu doświadczeniu	156
To zależy	157