

Spis treści

Przedmowa	ix
1. Rozpoczynanie podróży	i
Tematy	1
Funkcje języka Java	1
Projekt i architektura oprogramowania	2
Zasady SOLID	2
Testowanie	2
Podsumowanie rozdziałów	3
Utrwalenie wiedzy	4
2. Analizator wyciągów bankowych	5
Wyzwanie	5
Cel	5
Wymagania wobec projektu analizatora wyciągów bankowych	6
Zasada KISS	6
Zmienne	8
Łatwość utrzymania i antywzorce	9
Boska klasa	9
Powielanie kodu	9
Zasada pojedynczej odpowiedzialności	10
Spójność	14
Spójność na poziomie klas	16
Spójność na poziomie metod	20
Zależność	20
Testowanie	22
Testowanie automatyczne	22
Korzystanie z JUnit	23
Pokrycie kodu	26
Wnioski	27
Utrwalenie wiedzy	27
Podsumowanie wyzwania	27

3. Rozszerzanie analizatora wyciągów bankowych	29
Wyzwanie	29
Cel	29
Wymagania wobec rozszerzonego analizatora wyciągów bankowych	30
Zasada otwarte-zamknięte	30
Tworzenie wystąpienia interfejsu funkcjonalnego	33
Wyrażenia lambda	34
Pułapki interfejsów	34
Boski interfejs	34
Zbyt szczegółowy interfejs	35
API jawne kontra niejawne	36
Klasy domenowe czy wartości pierwotne?	38
Wiele eksporterów	38
Wprowadzenie obiektu domenowego	39
Definiowanie i implementowanie odpowiedniego interfejsu	40
Obsługa wyjątków	42
Dlaczego warto używać wyjątków?	42
Wzorce i antywzorce stosowania wyjątków	43
Wytyczne dotyczące stosowania wyjątków	48
Alternatywy dla wyjątków	49
Stosowanie narzędzia kompilacji	51
Dlaczego warto używać narzędzi kompilacji?	51
Korzystanie z Maven	52
Korzystanie z Gradle	55
Wnioski	57
Utrwalenie wiedzy	57
Podsumowanie wyzwania	58
4. System zarządzania dokumentami	59
Wyzwanie	59
Cel	59
Wymagania wobec systemu zarządzania dokumentami	60
Rozwijanie projektu	60
Importery	61
Klasa Document	62
Atrybuty i hierarchia dokumentów	64
Implementowanie i rejestrowanie importerów	65
Zasada podstawienia Liskov (LSP)	67
Alternatywne podejścia	69
Zastosowanie klasy Importer	69
Opcje związane z zasięgiem i hermetyzacją	69
Rozszerzanie i ponowne używanie kodu	70
Higiena testów	75
Nazewnictwo testów	75
Zachowanie, a nie implementacja	77

Nie powtarzaj się	78
Dobra diagnostyka	79
Testowanie błędów	81
Stałe	82
Wnioski	82
Utrwalenie wiedzy	83
Podsumowanie wyzwania	83
5. Silnik reguł biznesowych	85
Wyzwanie	85
Cel	85
Wymagania wobec silnika reguł biznesowych	86
Programowanie sterowane testami	86
Dlaczego warto używać techniki TDD?	87
Cykl TDD	88
Stosowanie atrap	90
Dodawanie warunków	91
Modelowanie stanu	91
Wnioskowanie typu zmiennej lokalnej	94
Wyrażenia switch	96
Zasada segregacji interfejsów	98
Projektowanie Fluent API	101
Czym jest Fluent API?	101
Modelowanie domeny	101
Wzorzec Budowniczy	103
Wnioski	105
Utrwalenie wiedzy	106
Podsumowanie wyzwania	106
6. Tootr	107
Wyzwanie	107
Cel	107
Wymagania wobec systemu Tootr	108
Przegląd projektu	109
Komunikacja typu pull	109
Komunikacja typu push	110
Od zdarzeń do projektu	111
Komunikacja	111
Graficzny interfejs użytkownika	112
Utrwalanie	112
Architektura sześciokątna	113
Od czego zacząć	114
Hasła i bezpieczeństwo	118
Obserwatorzy i tooty	120
Modelowanie błędów	121

Trootowanie	123
Tworzenie atrap	124
Weryfikowanie przy użyciu atrap obiektów	125
Biblioteki do stosowania atrap obiektów	126
SenderEndPoint	126
Pozycje	128
Metody equals i hashCode	131
Kontrakt między equals a hashCode	132
Wnioski	133
Utrwalenie wiedzy	133
Podsumowanie wyzwania	133
7. Rozszerzanie Trootra	135
Wyzwanie	135
Cel	135
Przypomnienie	135
Utrwalanie i wzorzec Repozytorium	136
Projektowanie repozytoriów	137
Obiekty kwerend	139
Programowanie funkcyjne	142
Wyrażenia lambda	143
Odwołania do metod	145
Execute Around	146
Strumienie	147
Optional	151
Interfejs użytkownika	154
Odwrócenie zależności i wstrzykiwanie zależności	154
Pakiety i systemy kompilacji	157
Ograniczenia i uproszczenia	159
Wnioski	160
Utrwalenie wiedzy	160
Podsumowanie wyzwania	160
8. Podsumowanie	161
Struktura bazująca na projekcie	161
Utrwalenie wiedzy	161
Świadoma praktyka	162
Kolejne kroki i dodatkowe zasoby	163
Indeks	165
0 autorach	177
0 ilustracji na okładce	179