

Słowo wstępne	11
Przedmowa	13
1. Czysty kod	17
1.1. Czym jest brzydki zapach kodu?	17
1.2. Co to jest refaktoryzacja?	17
1.3. Co to jest receptura?	18
1.4. Po co czysty kod?	18
1.5. Czytelność, wydajność czy jedno i drugie	19
1.6. Typy oprogramowania	19
1.7. Kod generowany maszynowo	19
1.8. Kwestie nazewnictwa	20
1.9. Wzorce projektowe	20
1.10. Paradygmaty języków oprogramowania	20
1.11. Obiekty kontra klasy	21
1.12. Łatwość wprowadzania zmian	21
2. Aksjomaty	22
2.0. Wprowadzenie	22
2.1. Dlaczego model?	23
2.2. Dlaczego abstrakcyjne?	23
2.3. Dlaczego programowalne?	23
2.4. Dlaczego częściowe?	24
2.5. Dlaczego wyjaśnienie?	24
2.6. Dlaczego rzeczywistości?	24
2.7. Wyprowadzanie reguł	24
2.8. Jedna i jedyna zasada projektowania oprogramowania	25

3. Modele anemiczne	30
3.0. Wprowadzenie	30
3.1. Przekształcanie obiektów anemicznych we wzbogacone	31
3.2. Identyfikowanie istoty obiektów	32
3.3. Usuwanie metod ustawiających	33
3.4. Usuwanie generatorów anemicznego kodu	35
3.5. Usuwanie właściwości automatycznych	37
3.6. Usuwanie obiektów DTO	38
3.7. Uzupełnianie pustych konstruktorów	41
3.8. Usuwanie metod pobierających	42
3.9. Zapobieganie orgiom obiektów	44
3.10. Usuwanie właściwości dynamicznych	46
4. Obsesja na punkcie prymitywów	48
4.0. Wprowadzenie	48
4.1. Tworzenie małych obiektów	49
4.2. Reifikacja danych prymitywnych	50
4.3. Reifikacja tablic asocjacyjnych	51
4.4. Usuwanie nadużyć łańcuchów	53
4.5. Reifikacja znaczników czasowych	54
4.6. Reifikacja podzbiorów jako obiektów	55
4.7. Reifikacja walidacji łańcuchów	56
4.8. Usuwanie niepotrzebnych właściwości	59
4.9. Tworzenie interwałów dat	60
5. Mutowalność	62
5.0. Wprowadzenie	62
5.1. Zmienianie deklaracji var na const	63
5.2. Deklarowanie zmiennych jako zmiennych	65
5.3. Zabranianie zmian istoty	66
5.4. Unikanie mutowalnych tablic stałych	67
5.5. Usuwanie leniwej inicjalizacji	69
5.6. Zamrażanie mutowalnych stałych	71
5.7. Usuwanie skutków ubocznych	72
5.8. Zapobieganie windowaniu	74
6. Kod deklaracyjny	75
6.0. Wprowadzenie	75
6.1. Zawężanie wielokrotnie wykorzystywanych zmiennych	76
6.2. Usuwanie pustych wierszy	77
6.3. Usuwanie wersjonowanych metod	78

6.4. Usuwanie podwójnych zaprzeczeń	79
6.5. Zmienianie źle przypisanych obowiązków	80
6.6. Zastępowanie jawnych iteracji	81
6.7. Dokumentowanie decyzji projektowych	82
6.8. Zastępowanie magicznych liczb stałymi	83
6.9. Oddzielanie „co” od „jak”	84
6.10. Dokumentowanie wyrażeń regularnych	85
6.11. Przekształcanie warunków Yody	86
6.12. Usuwanie dowcipnych metod	87
6.13. Unikanie piekła wywołań zwrotnych	88
6.14. Generowanie dobrych komunikatów o błędach	89
6.15. Unikanie magicznych poprawek	91
7. Nazewnictwo	93
7.0. Wprowadzenie	93
7.1. Rozwijanie skrótów	93
7.2. Zmienianie nazw oraz dzielenie klas pomocniczych i narzędziowych	95
7.3. Zmienianie nazw MoichObiektów	98
7.4. Zmienianie nazw zmiennych wynikowych	98
7.5. Zmienianie nazw pochodzących od typów	100
7.6. Zmienianie długich nazw	101
7.7. Zmienianie abstrakcyjnych nazw	102
7.8. Poprawianie pomyłek w pisowni	103
7.9. Usuwanie nazw klas z atrybutów	103
7.10. Usuwanie pierwszej litery z nazw klas i interfejsów	104
7.11. Zmienianie nazw funkcji zawierających słowa Basic/Do	105
7.12. Przekształcanie nazw klas w liczbie mnogiej w liczbę pojedynczą	107
7.13. Usuwanie z nazw słowa „Collection”	107
7.14. Usuwanie z nazw klas przedrostka/przyrostka „Impl”	108
7.15. Zmienianie nazw argumentów według roli	109
7.16. Usuwanie redundantnych nazw parametrów	110
7.17. Usuwanie z nazw bezcelowego kontekstu	111
7.18. Unikanie nazywania obiektów „danymi”	112
8. Komentarze	113
8.0. Wprowadzenie	113
8.1. Usuwanie „wykomentowanego” kodu	113
8.2. Usuwanie przestarzałych komentarzy	115
8.3. Usuwanie komentarzy logicznych	116
8.4. Usuwanie komentarzy do metod pobierających	118

8.5. Przekształcanie komentarzy w nazwy funkcji	119
8.6. Usuwanie komentarzy wewnątrz metod	119
8.7. Zastępowanie komentarzy testami	121
9. Standardy	124
9.0. Wprowadzenie	124
9.1. Przestrzeganie standardów kodu	124
9.2. Standaryzowanie wcięć	127
9.3. Unifikowanie wielkości liter	127
9.4. Pisanie kodu po angielsku	128
9.5. Unifikowanie kolejności parametrów	130
9.6. Naprawianie wybitych szyb	131
10. Złożoność	133
10.0. Wprowadzenie	133
10.1. Usuwanie powtarzalnego kodu	133
10.2. Usuwanie ustawień/konfiguracji i przełączników funkcji	135
10.3. Zmienianie stanu jako właściwości	136
10.4. Usuwanie z kodu pomysłowości	139
10.5. Łączenie wielu obietnic	140
10.6. Przerywanie długich łańcuchów współpracy	141
10.7. Wyodrębnianie metody do obiektu	142
10.8. Dbanie o konstruktory tablic	144
10.9. Usuwanie poltergeistów	145
11. Rozdęcie	147
11.0. Wprowadzenie	147
11.1. Dzielenie zbyt długich metod	147
11.2. Ograniczanie nadmiaru argumentów	148
11.3. Ograniczanie nadmiaru zmiennych	150
11.4. Usuwanie nadmiaru nawiasów	152
11.5. Usuwanie nadmiaru metod	153
11.6. Dzielenie zbyt licznych atrybutów	154
11.7. Ograniczanie list importu	155
11.8. Dzielenie funkcji wielozadaniowych	156
11.9. Dzielenie grubych interfejsów	158
12. YAGNI	160
12.0. Wprowadzenie	160
12.1. Usuwanie martwego kodu	160
12.2. Używanie kodu zamiast diagramów	162

12.3. Refaktoryzowanie klas z jedną podklasą	163
12.4. Usuwanie interfejsów jednorazowych	165
12.5. Usuwanie nadużyć wzorców projektowych	166
12.6. Zastępowanie kolekcji biznesowych	167
13. Szybkie reagowanie na błędy	170
13.0. Wprowadzenie	170
13.1. Refaktoryzowanie ponownego przypisywania zmiennych	170
13.2. Egzekwowanie warunków wstępnych	172
13.3. Używanie bardziej rygorystycznych parametrów	174
13.4. Usuwanie wartości domyślnej z instrukcji switch	175
13.5. Unikanie modyfikowania kolekcji podczas ich przetwarzania	176
13.6. Redefiniowanie haszowania i równości	178
13.7. Refaktoryzacja bez zmian funkcjonalnych	179
14. Instrukcje if	181
14.0. Wprowadzenie	181
14.1. Zastępowanie akcydentalnych instrukcji if polimorfizmem	182
14.2. Zmienianie nazw zmiennych sygnalizujących zdarzenia	188
14.3. Reifikacja zmiennych logicznych	189
14.4. Zastępowanie instrukcji switch/case/elseif	190
14.5. Zastępowanie zakodowanych „na sztywno” warunków if kolekcjami	192
14.6. Zmienianie warunków logicznych na zwarciove	193
14.7. Dodawanie niejawnego warunku else	194
14.8. Poprawianie warunkowego kodu strzałkowego	195
14.9. Unikanie hakerskich zwarć	197
14.10. Poprawianie zagnieżdżonego kodu strzałkowego	198
14.11. Zapobieganie zwracaniu wartości logicznych podczas sprawdzania warunków	199
14.12. Zmienianie porównań z wartościami logicznymi	201
14.13. Wyodrębnianie kodu z długich warunków trójargumentowych	202
14.14. Przekształcanie funkcji niepolimorficznych w polimorficzne	203
14.15. Zmienianie sprawdzania równości	205
14.16. Reifikacja zakodowanych „na sztywno” warunków biznesowych	206
14.17. Usuwanie nieuzasadnionych wartości logicznych	207
14.18. Poprawianie zagnieżdżonych warunków trójargumentowych	208
15. Null	210
15.0. Wprowadzenie	210
15.1. Tworzenie obiektów null	210
15.2. Usuwanie opcjonalnego dostępu łańcuchowego	213

15.3. Przekształcanie opcjonalnych atrybutów w kolekcję	215
15.4. Używanie prawdziwych obiektów do reprezentowania wartości null	217
15.5. Reprezentowanie nieznanego lokacji bez użycia null	220
16. Przedwczesna optymalizacja	223
16.0. Wprowadzenie	223
16.1. Unikanie identyfikatorów w obiektach	224
16.2. Usuwanie przedwczesnej optymalizacji	226
16.3. Usuwanie przedwczesnej optymalizacji w postaci operacji bitowych	228
16.4. Ograniczanie nadmiernej generalizacji	229
16.5. Zmienianie optymalizacji strukturalnych	230
16.6. Usuwanie zakotwiczonych łodzi	231
16.7. Wyodrębnianie pamięci podręcznych z obiektów dziedzicznych	232
16.8. Usuwanie zdarzeń opartych na implementacji	234
16.9. Usuwanie kwerend z konstruktorów	235
16.10. Usuwanie kodu z destruktorów	236
17. Sprzężenie	239
17.0. Wprowadzenie	239
17.1. Ujawnianie ukrytych założeń	239
17.2. Zastępowanie singletonów	241
17.3. Rozbijanie boskich obiektów	242
17.4. Rozdzielanie rozbieżnych zmian	245
17.5. Przekształcanie specjalnych flag typu 9999 w normalne wartości	246
17.6. Usuwanie chirurgii strzelbowej	248
17.7. Usuwanie argumentów opcjonalnych	250
17.8. Zapobieganie zawiści funkcjonalnej	251
17.9. Usuwanie pośrednika	253
17.10. Przenoszenie argumentów domyślnych na koniec	254
17.11. Unikanie efektu domina	255
17.12. Usuwanie akcydentalnych metod z obiektów biznesowych	257
17.13. Usuwanie kodu biznesowego z interfejsu użytkownika	258
17.14. Zmienianie sprzężenia z klasami	261
17.15. Refaktoryzacja zbitek danych	262
17.16. Przerywanie nieestosownej bliskości	264
17.17. Przekształcanie obiektów wymiennych	266
18. Globalność	268
18.0. Wprowadzenie	268
18.1. Reifikacja funkcji globalnych	268
18.2. Reifikacja funkcji statycznych	269

18.3. Zastępowanie instrukcji goto ustrukturyzowanym kodem	270
18.4. Usuwanie klas globalnych	271
18.5. Zmianianie globalnego tworzenia dat	273
19. Hierarchie	275
19.0. Wprowadzenie	275
19.1. Eliminowanie głębokiego dziedziczenia	275
19.2. Rozbijanie hierarchii jojo	278
19.3. Eliminowanie subklasyfikacji stosowanej w celu wielokrotnego wykorzystywania kodu	279
19.4. Zastępowanie relacji „jest” działaniem	281
19.5. Usuwanie zagnieżdżonych klas	283
19.6. Zmianianie nazw izolowanych klas	284
19.7. Oznaczanie klas konkretnych jako finalnych	285
19.8. Jawne definiowanie dziedziczenia klas	287
19.9. Migrowanie pustych klas	288
19.10. Opóźnianie przedwczesnej optymalizacji	290
19.11. Usuwanie atrybutów chronionych	291
19.12. Uzupełnianie pustych implementacji	293
20. Testowanie	295
20.0. Wprowadzenie	295
20.1. Testowanie metod prywatnych	295
20.2. Dodawanie opisów do asercji	297
20.3. Migrowanie assertTrue do konkretnych asercji	298
20.4. Zastępowanie atrap prawdziwymi obiektami	300
20.5. Dopracowywanie ogólnikowych asercji	301
20.6. Usuwanie niestabilnych testów	302
20.7. Zmianianie asercji z liczbami zmiennoprzecinkowymi	304
20.8. Zmianianie danych testowych w dane realistyczne	305
20.9. Eliminowanie testów naruszających enkapsulację	307
20.10. Usuwanie nieistotnych informacji testowych	309
20.11. Dodawanie pokrycia testowego do każdego wniosku o scalenie	310
20.12. Poprawianie testów zależnych od dat	311
20.13. Nauka nowego języka programowania	313
21. Dług techniczny	314
21.0. Wprowadzenie	314
21.1. Usuwanie kodu zależnego od środowiska produkcyjnego	315
21.2. Usuwanie list znanych defektów	316
21.3. Włączanie ostrzeżeń/ rygorystycznej kontroli błędów	317
21.4. Unikanie i usuwanie rzeczy „do zrobienia” i „do poprawki”	319

22. Wyjątki	321
22.0. Wprowadzenie	321
22.1. Usuwanie pustych bloków wyjątków	321
22.2. Usuwanie niepotrzebnych wyjątków	322
22.3. Poprawianie wyjątków obsługujących oczekiwane przypadki	324
22.4. Poprawianie zagnieżdżonych instrukcji try/catch	325
22.5. Zastępowanie kodów zwrotnych wyjątkami	326
22.6. Poprawianie kodu strzałkowego do obsługi wyjątków	328
22.7. Ukrywanie niskopoziomowych błędów przed użytkownikami końcowymi	329
22.8. Zawężanie bloków try w wyjątkach	330
23. Metaprogramowanie	332
23.0. Wprowadzenie	332
23.1. Usuwanie użycia metaprogramowania	332
23.2. Reifikacja funkcji anonimowych	336
23.3. Usuwanie preprocesorów	338
23.4. Usuwanie metod dynamicznych	339
24. Typy	341
24.0. Wprowadzenie	341
24.1. Usuwanie sprawdzania typów	341
24.2. Radzenie sobie z wartościami pseudoprawdziwymi	343
24.3. Zmienianie liczb zmiennoprzecinkowych w dziesiętne	345
25. Bezpieczeństwo	348
25.0. Wprowadzenie	348
25.1. Sanityzacja danych wejściowych	348
25.2. Zmienianie sekwencyjnych identyfikatorów	350
25.3. Usuwanie zależności od pakietów	351
25.4. Zastępowanie szkodliwych wyrażeń regularnych	352
25.5. Ochrona deserializacji obiektu	354
Słownik terminów	356