

Spis treści

Wstęp	11
1. Sztuka projektowania oprogramowania	17
Wytyczna 1.: Znaczenie projektu oprogramowania	18
Cechy to nie projekt oprogramowania	18
Projektowanie oprogramowania: Sztuka zarządzania zależnościami i abstrakcjami	19
Trzy poziomy projektowania oprogramowania	21
Zwracanie uwagi na możliwości	25
Zwracanie uwagi na projekt oprogramowania oraz zasady projektowe	26
Wytyczna 2.: Projektuj pod kątem zmian	27
Separacja zagadnień	27
Przykład sztucznych powiązań	28
Powiązania logiczne oraz fizyczne	31
Nie powtarzaj się	34
Unikaj zbyt wczesnego separowania zagadnień	37
Wytyczna 3.: Separuj interfejsy w celu unikania sztucznych powiązań	39
Segregacja interfejsów w celu separacji zagadnień	39
Minimalizacja wymagań określanych przez argumenty szablonów	41
Wytyczna 4.: Projektuj pod kątem łatwości testowania	42
Jak testować prywatną funkcję składową?	43
Prawdziwe rozwiązanie: Separacja zagadnień	47
Wytyczna 5.: Projektuj pod kątem rozszerzania	49
Zasada otwarte-zamknięte	50
Rozszerzalność podczas kompilacji	54
Unikanie przedwczesnego projektowania pod kątem rozszerzania	56
2. Sztuka tworzenia abstrakcji	58
Wytyczna 6.: Trzymaj się oczekiwanych zachowań abstrakcji	59
Przykład naruszania oczekiwań	59
Zasada podstawienia Liskov	61
Krytyka zasady podstawienia Liskov	65
Potrzeba dobrych i sensownych abstrakcji	66
Wytyczna 7.: Zrozum podobieństwa pomiędzy klasami bazowymi a konceptami	66
Wytyczna 8.: Zrozum semantyczne wymagania zbiorów przeciążeń	70
Potęga funkcji zewnętrznych: mechanizm abstrakcji czasu kompilacji	70
Problem funkcji zewnętrznych: Oczekiwane zachowanie	73
Wytyczna 9.: Zwracaj uwagę na własność abstrakcji	76
Zasada odwrócenia zależności	76

Odwrócenie zależności w architekturze opartej na wtyczkach	82
Odwrócenie zależności z wykorzystaniem szablonów	84
Odwrócenie zależności z wykorzystaniem zbioru przeciążeń	85
Zasada odwrócenia zależności kontra zasada jednej odpowiedzialności	86
Wtyczna 10.: Rozważ stworzenie dokumentacji architektury	87
3. Przeznaczenie wzorców projektowych	91
Wtyczna 11.: Zrozum przeznaczenie wzorców projektowych	92
Wzorzec projektowy ma nazwę	92
Wzorce projektowe mają swoje przeznaczenie	93
Wzorce projektowe wprowadzają abstrakcję	94
Przydatność wzorca potwierdzono w praktyce	96
Wtyczna 12.: Strzeż się błędnych przekonań dotyczących wzorców projektowych	97
Wzorce projektowe nie są celem	97
We wzorcach projektowych nie chodzi o szczegóły implementacyjne	98
Wzorce projektowe nie ograniczają się do języków programowania zorientowanych obiektowo ani do polimorfizmu dynamicznego	101
Wtyczna 13.: Wzorce projektowe są wszędzie	104
Wtyczna 14.: Używaj nazwy wzorca, by wyrazić jego przeznaczenie	108
4. Wzorzec projektowy Odwiedzający	111
Wtyczna 15.: Projektuj pod kątem dodawania typów i operacji	112
Rozwiązanie proceduralne	112
Rozwiązanie obiektowe	118
Uważaj na decyzję projektową związaną z polimorfizmem dynamicznym	120
Wtyczna 16.: Stosowanie wzorca Odwiedzający do rozszerzania operacji	122
Analiza problemów z projektem	122
Prezentacja wzorca projektowego Odwiedzający	123
Analiza wad wzorca projektowego Odwiedzający	127
Wtyczna 17.: Rozważ użycie std::variant do implementacji wzorca Odwiedzający	130
Wprowadzenie do std::variant	131
Refaktoryzacja rysowania figur z użyciem nieintryzyjnego rozwiązania opartego na wartościach	133
Pomiary wydajności działania	138
Analiza wad rozwiązania korzystającego z std::variant	140
Wtyczna 18.: Uważaj na wydajność acyklicznego odwiedzającego	141
5. Wzorce projektowe Strategia i Polecenie	147
Wtyczna 19.: Stosuj wzorzec Strategia do określania sposobu wykonywania operacji	148
Analiza wad projektu	150
Prezentowanie wzorca projektowego Strategia	153
Analiza mankamentów naiwnej implementacji Strategii	157
Porównanie wzorców Odwiedzający i Strategia	162

Analiza mankamentów wzorca projektowego Strategia	163
Projekt oparty na strategii	165
Wytyczna 20.: Przedkładaj kompozycję nad dziedziczenie	168
Wytyczna 21.: Stosuj wzorzec Polecenie, by izolować operacje do wykonania	171
Prezentacja wzorca projektowego Polecenie	171
Porównanie wzorców projektowych Polecenie i Strategia	178
Analiza mankamentów wzorca projektowego Polecenie	180
Wytyczna 22.: Przedkładaj semantykę wartości nad semantykę referencji	181
Wady stylu z książki Bandy Czworga: semantyka referencji	181
Semantyka referencji: drugi przykład	185
Filozofia nowoczesnego C++: semantyka wartości	187
Semantyka wartości: drugi przykład	188
W implementacjach wzorców projektowych preferuj stosowanie semantyki wartości	191
Wytyczna 23.: Preferuj implementację wzorca Strategia korzystającą z wartości	191
Prezentowanie std::function	191
Refaktoryzacja rysowania figur	193
Pomiary wydajności działania	197
Analiza mankamentów rozwiązania korzystającego z std::function	199
6. Wzorce projektowe: Adapter, Obserwator i CRTP	201
Wytyczna 24.: Stosuj adaptory, by standaryzować interfejsy	202
Prezentacja wzorca projektowego Adapter	203
Adaptory obiektowe a adaptory klasowe	205
Przykłady z Biblioteki standardowej	206
Porównanie wzorców Adapter i Strategia	208
Adaptory funkcyjne	208
Analiza mankamentów wzorca projektowego Adapter	210
Wytyczna 25.: Stosuj wzorzec Obserwator jako abstrakcyjny mechanizm powiadamiania	213
Prezentacja wzorca projektowego Obserwator	213
Klasyczna implementacja wzorca projektowego Obserwator	214
Implementacja obserwatora oparta na semantyce wartości	224
Analiza mankamentów wzorca projektowego Obserwator	226
Wytyczna 26.: Stosuj wzorzec CRTP, by wprowadzać statyczne kategorie typów	227
Przeznaczenie wzorca CRTP	228
Prezentacja wzorca projektowego CRTP	233
Analiza mankamentów wzorca projektowego CRTP	238
Przyszłość wzorca projektowego CRTP: Porównanie wzorca CRTP i konceptów C++20	240
Wytyczna 27.: Stosuj wzorzec CRTP do tworzenia statycznych klas domieszek	243
Chęć posiadania silnego typu	243
Stosowanie CRTP jako wzorca implementacyjnego	245

7. Wzorce projektowe Most, Prototyp oraz Polimorfizm zewnętrzny	250
Wytyczna 28.: Tworzenie mostów w celu wyeliminowania fizycznych zależności	251
Przykład motywujący	251
Opis wzorca projektowego Most	255
Idiom Pimpl	258
Porównanie wzorców projektowych Most i Strategia	262
Analiza mankamentów wzorca projektowego Most	264
Wytyczna 29.: Bądź świadom zysków i strat wydajności we wzorcu projektowym Most	266
Wpływ użycia wzorca projektowego Most na wydajność działania	266
Poprawianie wydajności przez zastosowanie wzorca projektowego Most	269
Wytyczna 30.: Stosuj wzorzec Prototyp, by wyodrębnić operacje kopiowania	271
Przykład zootechniczny: Kopiowanie zwierząt	272
Prezentacja wzorca projektowego Prototyp	273
Porównanie wzorca projektowego Prototyp i szablonu klasy std::variant	276
Analiza mankamentów wzorca projektowego Prototyp	277
Wytyczna 31.: Stosuj wzorzec Polimorfizm zewnętrzny, by tworzyć nieintruzyjny polimorfizm czasu wykonywania	278
Prezentacja wzorca projektowego Polimorfizm zewnętrzny	279
Rysowanie figur raz jeszcze	282
Porównanie wzorców projektowych Polimorfizm zewnętrzny i Adapter	289
Analiza mankamentów wzorca projektowego Polimorfizm zewnętrzny	289
8. Wzorzec projektowy Ukrywanie typu	293
Wytyczna 32.: Rozważ zastąpienie hierarchii dziedziczenia wzorcem projektowym Ukrywanie typu	294
Historia ukrywania typu	294
Prezentacja wzorca projektowego Ukrywanie typu	297
Implementacja wzorca projektowego Ukrywanie typu mająca prawa własności	298
Analiza mankamentów wzorca projektowego Ukrywanie typu	306
Porównanie dwóch rodzajów opakowań stosowanych we wzorcu Ukrywanie typu	307
Segregacja interfejsów w opakowaniach wzorca projektowego Ukrywanie typu	309
Wyniki pomiarów wydajności	310
Kilka słów o terminologii	312
Wytyczna 33.: Miej świadomość optymalizacyjnego potencjału wzorca projektowego Ukrywanie typu	313
Optymalizacja małego bufora	314
Ręczna implementacja przydzielania funkcji	322
Wytyczna 34.: Pamiętaj o kosztach konfiguracji związanych z rodzajem opakowań używanych we wzorcu	327

Ukrywanie typu	327
Koszty konfiguracji związane z rodzajem opakowań stosowanych we wzorcu Ukrywanie typu	327
Prosta implementacja wzorca projektowego Ukrywanie typu niemająca praw własności	329
Pozbawiona praw własności implementacja Ukrywania typu o większych możliwościach	331
9. Wzorzec projektowy Dekorator	340
Wytyczna 35.: Stosuj dekoratory, aby dodawać dostosowania hierarchicznie	340
Problem projektowy kolegów z firmy	341
Prezentacja wzorca projektowego Dekorator	345
Klasyczna implementacja wzorca projektowego Dekorator	347
Drugi przykład dekoratora	351
Porównanie wzorców projektowych Dekorator, Adapter i Strategia	354
Analiza mankamentów wzorca projektowego Dekorator	355
Wytyczna 36.: Zrozum kompromis pomiędzy abstrakcją czasu wykonywania a abstrakcją czasu kompilacji	358
Dekorator czasu kompilacji oparty na wartościach	358
Wartościowy dekorator czasu wykonania	363
10. Wzorzec projektowy Singleton	368
Wytyczna 37.: Traktuj Singleton jako wzorzec implementacyjny, a nie wzorzec projektowy	368
Przedstawienie wzorca Singleton	369
Singleton nie zarządza zależnościami ani ich nie redukuje	372
Wytyczna 38.: Projektuj singletony pod kątem zmian i możliwości testowania	374
Singletony reprezentują globalny stan	374
Singletony utrudniają wprowadzanie zmian i możliwości testowania	376
Odwrócenie zależności od singletonu	379
Stosowanie wzorca projektowego Strategia	383
Podążając w kierunku wstrzykiwania zależności	388
11. Ostatnia wytyczna	391
Wytyczna 39.: Kontynuuj poznawanie wzorców projektowych	391