

Projektowanie frameworków w .NET : wytyczne, konwencje idiomy i wzorce / Krzysztof Cwalina, Jeremy Barton, Brad Abrams. – Gliwice, 2021

Spis treści

Spis rysunków	13
Spis tabel	15
Przedmowa	17
Przedmowa do wydania drugiego	19
Przedmowa do wydania pierwszego	21
Wstęp	23
Podziękowania	27
O autorach	29
O komentatorach	31
1. Wprowadzenie	37
1.1. Walory dobrze zaprojektowanego frameworka	38
1.1.1. Dobrze zaprojektowane frameworki są proste	38
1.1.2. Dobrze zaprojektowane frameworki muszą kosztować	39
1.1.3. Dobrze zaprojektowane frameworki są pełne kompromisów	40
1.1.4. Dobrze zaprojektowane frameworki zawierają zapożyczenia z przeszłości	41
1.1.5. Dobrze zaprojektowane frameworki można rozwijać	41
1.1.6. Dobrze zaprojektowane frameworki można integrować	42
1.1.7. Dobrze zaprojektowane frameworki są spójne	42
2. Podstawy projektowania frameworków	43
2.1. Progresywne frameworki	45
2.2. Podstawowe zasady projektowania frameworków	48
2.2.1. Zasada projektowania opartego na scenariuszach	48
2.2.2. Zasada niskiego progu wejścia	54
2.2.3. Zasada samodokumentujących się modeli obiektów	59
2.2.4. Zasada architektury warstwowej	64
Podsumowanie	66
3. Wytyczne dla nazw	67
3.1. Konwencje dotyczące wielkości liter	67

3.1.1. Reguły stosowania wielkich liter w identyfikatorach	68
3.1.2. Wielkie litery w akronimach	70
3.1.3. Wielkie litery w wyrazach złożonych i często używanych zwrotach	72
3.1.4. Rozróżnianie wielkości znaków	74
3.2. Ogólne konwencje nazewnictwa	75
3.2.1. Dobór słów	75
3.2.2. Używanie skrótów i akronimów	77
3.2.3. Unikanie nazw specyficznych dla języków	78
3.2.4. Nazewnictwo nowych wersji istniejących interfejsów API	79
3.3. Nazwy zestawów, bibliotek DLL i pakietów	81
3.4. Nazwy przestrzeni nazw	83
3.4.1. Przestrzeń nazw i konflikty nazw typów	84
3.5. Nazwy klas, struktur i interfejsów	86
3.5.1. Nazwy uogólnionych parametrów typowych	88
3.5.2. Nazwy zwykłych typów	89
3.5.3. Nazewnictwo wyliczeń	90
3.6. Nazwy składowych typów	91
3.6.1. Nazwy metod	91
3.6.2. Nazwy właściwości	92
3.6.3. Nazwy zdarzeń	93
3.6.4. Nazwy pól	94
3.7. Nazwy parametrów	95
3.7.1. Nazwy parametrów przeciążonych operatorów	95
3.8. Nazewnictwo zasobów	96
Podsumowanie	96
4. Wytyczne dotyczące projektowania typów	97
4.1. Typy i przestrzeń nazw	99
4.2. Wybór między klasą a strukturą	102
4.3. Wybór między klasą a interfejsem	105
4.4. Projektowanie klas abstrakcyjnych	111
4.5. Projektowanie klas statycznych	112
4.6. Projektowanie interfejsów	113
4.7. Projektowanie struktur	115
4.8. Projektowanie wyliczeń	119
4.8.1. Projektowanie wyliczeń znacznikowych	125
4.8.2. Dodawanie wartości do wyliczeń	128
4.9. Typy zagnieżdżone	130
4.10. Typy a metadane zestawów	132
4.11. Silnie typowane ciągi	133
Podsumowanie	136
5. Projektowanie składowych	137
5.1. Ogólne wytyczne dotyczące projektowania składowych	137
5.1.1. Przeciążanie składowych	137
5.1.2. Jawne implementowanie składowych interfejsu	147
5.1.3. Wybór między właściwościami a metodami	150

5.2. Projektowanie właściwości	154
5.2.1. Projektowanie właściwości indeksowanych	157
5.2.2. Zdarzenia powiadamiania o zmianach właściwości	159
5.3. Projektowanie konstruktorów	160
5.3.1. Wytyczne dla konstruktorów typów	166
5.4. Projektowanie zdarzeń	168
5.5. Projektowanie pól	172
5.6. Metody rozszerzające	175
5.7. Przeciążanie operatorów	181
5.7.1. Przeciążanie operatora	185
5.7.2. Operatory konwersji	185
5.7.3. Operatory nierówności	186
5.8. Projektowanie parametrów	188
5.8.1. Wybór między parametrami typu wyliczeniowego i logicznego	190
5.8.2. Sprawdzanie poprawności argumentów	192
5.8.3. Przekazywanie parametrów	195
5.8.4. Składowe ze zmienną liczbą parametrów	198
5.8.5. Parametry wskaźnikowe	201
5.9. Używanie krotek w sygnaturach składowych	202
Podsumowanie	207
6. Projektowanie pod kątem rozszerzalności	209
6.1. Mechanizmy rozszerzalności	209
6.1.1. Klasy niezapieczętowane	209
6.1.2. Składowe chronione	211
6.1.3. Zdarzenia i wywołania zwrotne	212
6.1.4. Składowe wirtualne	217
6.1.5. Abstrakcje (typy i interfejsy abstrakcyjne)	219
6.2. Klasy bazowe	220
6.3. Pieczętowanie	222
Podsumowanie	224
7. Wyjątki	225
7.1. Zgłaszanie wyjątków	229
7.2. Wybór odpowiedniego typu zgłaszanego wyjątku	234
7.2.1. Opracowywanie komunikatu o błędzie	236
7.2.2. Obsługa wyjątków	237
7.2.3. Opakowywanie wyjątków	242
7.3. Korzystanie ze standardowych typów wyjątków	244
7.3.1. Exception i SystemException	244
7.3.2. ApplicationException	244
7.3.3. InvalidOperationException	244
7.3.4. ArgumentException, ArgumentNullException i ArgumentOutOfRangeException	245
7.3.5. NullReferenceException, IndexOutOfRangeException i AccessViolationException	246
7.3.6. StackOverflowException	246

7.3.7. OutOfMemoryException	246
7.3.8. ComException, SEHException i ExecutionEngineException	247
7.3.9. OperationCanceledException i TaskCanceledException	247
7.3.10. FormatException	248
7.3.11. PlatformNotSupportedException	248
7.4. Projektowanie własnych wyjątków	248
7.5. Wyjątki a wydajność	249
7.5.1. Wzorzec Tester-Wykonawca	250
7.5.2. Wzorzec Try	251
Podsumowanie	254
8. Wytyczne dotyczące użytkowania	255
8.1. Tablice	255
8.2. Atrybuty	258
8.3. Kolekcje	260
8.3.1. Kolekcje jako parametry	262
8.3.2. Kolekcje jako właściwości i wartości zwracane	263
8.3.3. Wybór między tablicą a kolekcją	266
8.3.4. Implementowanie kolekcji niestandardowych	267
8.4. DateTime i DateTimeOffset	269
8.5. ICloneable	271
8.6. IComparable<T> i IEquatable<T>	271
8.7. IDisposable	273
8.8. Nullable<T>	273
8.9. Object	274
8.9.1. Object.Equals	274
8.9.2. Object.GetHashCode	276
8.9.3. Object.ToString	277
8.10. Serializacja	279
8.11. Uri	281
8.11.1. Wytyczne implementacyjne dotyczące typu System.Uri	282
8.12. Użycie przestrzeni nazw System.Xml	282
8.13. Operatory równości	283
8.13.1. Operatory równości w typach wartościowych	285
8.13.2. Operatory równości w typach referencyjnych	286
9. Typowe wzorce projektowe	287
9.1. Komponenty agregujące	287
9.1.1. Projektowanie komponentowe	289
9.1.2. Typy sfactoryzowane	291
9.1.3. Wytyczne dotyczące komponentów agregujących	292
9.2. Wzorce asynchroniczne	294
9.2.1. Wybór między wzorcami asynchronicznymi	295
9.2.2. Wzorzec asynchroniczny oparty na zadaniach	296
9.2.3. Typy zwracane z metod asynchronicznych	301
9.2.4. Tworzenie asynchronicznego wariantu istniejącej metody synchronicznej	304

9.2.5. Wytyczne implementacyjne mające na celu zachowanie spójności wzorca asynchronicznego	306
9.2.6. Klasyczny wzorzec asynchroniczny	311
9.2.7. Wzorzec asynchroniczny oparty na zdarzeniach	311
9.2.8. <code>IAsyncDisposable</code>	312
9.2.9. <code>IAsyncEnumerable<T></code>	312
9.3. Właściwości zależnościowe	314
9.3.1. Projektowanie właściwości zależnościowych	315
9.3.2. Projektowanie dołączanych właściwości zależnościowych	317
9.3.3. Sprawdzanie poprawności właściwości zależnościowych	318
9.3.4. Powiadomienia o zmianach właściwości zależnościowych	318
9.3.5. Koercja wartości właściwości zależnościowej	319
9.4. Wzorzec <code>Dispose</code>	320
9.4.1. Podstawowy wzorzec <code>Dispose</code>	322
9.4.2. Typy finalizowalne	328
9.4.3. Operacje z określonym zakresem	331
9.4.4. <code>IAsyncDisposable</code>	334
9.5. Fabryki	337
9.6. Obsługa LINQ	341
9.6.1. Omówienie mechanizmu LINQ	341
9.6.2. Sposoby implementowania obsługi technologii LINQ	342
9.6.3. Obsługa technologii LINQ za pośrednictwem interfejsu <code>IEnumerable<T></code>	342
9.6.4. Obsługa LINQ za pośrednictwem interfejsu <code>IQueryable<T></code>	343
9.6.5. Obsługa technologii LINQ za pośrednictwem wzorca <code>Query</code>	344
9.7. Wzorzec funkcji opcjonalnych	347
9.8. Kowariancja i kontrawariancja	350
9.8.1. Kontrawariancja	352
9.8.2. Kowariancja	354
9.8.3. Wzorzec symulowanej kowariancji	356
9.9. Metoda szablonowa	359
9.10. Limity czasu	361
9.11. Odczytywanie typów z kodu XAML	362
9.12. Operacje na buforach	364
9.12.1. Operacje transformacji danych	375
9.12.2. Zapisywanie w buforze danych o stałej lub wstępnie określonej wielkości	380
9.12.3. Zapisywanie w buforze danych z użyciem wzorca <code>Try-Write</code>	381
9.12.4. Częściowe zapisy do buforów i wyliczenie <code>OperationStatus</code>	385
9.13. A na koniec...	389
A Konwencje stylu programowania w C#	391
A.1. Ogólne konwencje stylu	392
A.1.1. Użycie nawiasów klamrowych	392
A.1.2. Użycie spacji	394
A.1.3. Użycie wcięć	395
A.1.4. Odstępy w pionie	397

A.1.5. Modyfikatory składowych	397
A.1.6. Inne	399
A.2. Konwencje nazewnictwa	403
A.3. Komentarze	404
A.4. Organizacja plików	405
B Przeszarzałe wytyczne	407
B.3. Przeszarzałe wytyczne dotyczące nazewnictwa	408
B.3.8. Nazewnictwo zasobów	408
B.4. Przeszarzałe wytyczne dotyczące projektowania typów	408
B.4.1. Typy i przestrzenie nazw	408
B.5. Przeszarzałe wytyczne dotyczące projektowania składowych	410
B.5.4. Projektowanie zdarzeń	410
B.7. Przeszarzałe wytyczne dotyczące wyjątków	411
B.7.4. Projektowanie wyjątków niestandardowych	411
B.8. Przeszarzałe wytyczne dotyczące użytkowania	412
B.8.10. Serializacja	412
B.9. Przeszarzałe wytyczne dotyczące typowych wzorców projektowych	419
B.9.2. Wzorce asynchroniczne	419
B.9.4. Wzorzec Dispose	429
C Przykład specyfikacji API	435
D Zmiany powodujące niezgodność	441
D.1. Modyfikowanie zestawów	442
D.1.1. Zmiana nazwy zestawu	442
D.2. Dodawanie przestrzeni nazw	443
D.2.1. Dodawanie przestrzeni nazw powodującej konflikt z istniejącym typem	443
D.3. Modyfikowanie przestrzeni nazw	443
D.3.1. Zmiana nazwy przestrzeni nazw lub wielkości jej liter	443
D.4. Przenoszenie typów	443
D.4.1. Przenoszenie typu za pośrednictwem atrybutu [TypeForwardedTo]	443
D.4.2. Przenoszenie typu bez użycia atrybutu [TypeForwardedTo]	444
D.5. Usuwanie typów	444
D.5.1. Usuwanie typów	444
D.6. Modyfikowanie typów	445
D.6.1. Zapieczętowanie typu niezapieczętowanego	445
D.6.2. Odpieczętowanie typu zapieczętowanego	445
D.6.3. Zmiana wielkości liter w nazwie typu	445
D.6.4. Zmiana nazwy typu	446
D.6.5. Zmiana przestrzeni nazw typu	446
D.6.6. Dodawanie do struktury modyfikatora readonly	446
D.6.7. Usuwanie ze struktury modyfikatora readonly	447
D.6.8. Dodawanie interfejsu bazowego do istniejącego interfejsu	447
D.6.9. Dodawanie drugiej deklaracji uogólnionego interfejsu	447
D.6.10. Zmiana klasy na strukturę	448

D.6.11. Zmiana struktury na klasę	448
D.6.12. Zmiana struktury na typ ref struct	449
D.6.13. Zmiana typu ref struct na strukturę (bez słowa kluczowego ref)	449
D.7. Dodawanie składowych	449
D.7.1. Maskowanie składowych typu bazowego za pomocą modyfikatora new	449
D.7.2. Dodawanie składowych abstrakcyjnych	450
D.7.3. Dodawanie składowych do typu niezapieczonego	450
D.7.4. Dodawanie składowej z modyfikatorem override do typu niezapieczonego	450
D.7.5. Dodawanie do struktury pierwszego pola typu referencyjnego	451
D.7.6. Dodawanie składowej do interfejsu	451
D.8. Przenoszenie składowych	452
D.8.1. Przenoszenie składowych do klasy bazowej	452
D.8.2. Przenoszenie składowych do interfejsu bazowego	452
D.8.3. Przenoszenie składowych do typu pochodnego	452
D.9. Usuwanie składowych	452
D.9.1. Usuwanie Analizatora z typu niezapieczonego	452
D.9.2. Usuwanie Analizatora z typu zapieczonego	453
D.9.3. Usuwanie składowej bez modyfikatora override	453
D.9.4. Usuwanie przesłonięcia składowej wirtualnej	453
D.9.5. Usuwanie przesłonięcia składowej abstrakcyjnej	454
D.9.6. Usuwanie lub zmiana nazwy pól prywatnych w typach serializowalnych	454
D.10. Przeciążanie składowych	454
D.10.1. Dodawanie pierwszego przeciążenia składowej	455
D.10.2. Dodawanie przeciążenia z alternatywnym parametrem typu referencyjnego	455
D.11. Zmiana sygnatur składowych	455
D.11.1. Zmiana nazwy parametru metody	455
D.11.2. Dodawanie lub usuwanie parametru metody	456
D.11.3. Zmiana typu parametru metody	456
D.11.4. Zmiana kolejności parametrów o różnych typach w metodzie	456
D.11.5. Zmiana kolejności parametrów tego samego typu w metodzie	457
D.11.6. Zmiana typu zwrotnego metody	457
D.11.7. Zmiana typu właściwości	458
D.11.8. Zmiana widoczności składowej z publicznej na dowolną inną	458
D.11.9. Zmiana widoczności składowej z chronionej na publiczną	458
D.11.10. Zmiana składowej wirtualnej (lub abstrakcyjnej) z chronionej na publiczną	458
D.11.11. Dodawanie lub usuwanie modyfikatora static	459
D.11.12. Rozpoczęcie przekazywania parametru przez referencję lub rezygnacja z tego	459
D.11.13. Zmiana stylu parametru referencyjnego	460
D.11.14. Nadawanie modyfikatora readonly metodzie struktury	460
D.11.15. Usuwanie modyfikatora readonly z metody struktury	460
D.11.16. Zmiana parametru z obowiązkowego na opcjonalny	460

D.11.17. Zmiana parametru z opcjonalnego na obowiązkowy	461
D.11.18. Zmiana wartości domyślnej parametru opcjonalnego	461
D.11.19. Zmiana wartości pola const	461
D.11.20. Zmiana składowej abstrakcyjnej na wirtualną	462
D.11.21. Zmiana składowej wirtualnej na abstrakcyjną	462
D.11.22. Zmiana składowej niewirtualnej na wirtualną	462
D.12. Zmiana działania	463
D.12.1. Zastępowanie wyjątków dotyczących błędów czasu wykonania wyjątkami odnoszącymi się do błędów użycia	463
D.12.2. Zastępowanie wyjątków dotyczących błędów użycia funkcjonalnym działaniem	463
D.12.3. Zmiana typu wartości zwracanych z metody	463
D.12.4. Zgłaszanie nowego typu wyjątku błędu	464
D.12.5. Zgłaszanie nowego typu wyjątku, odziedziczonego po dotychczas zgłaszanym typie	464
D.13. Ostatnia uwaga	464
Słowniczek	465

oprac. BPK