

Spis treści

O autorze	xiii
Podziękowania	xv
Przedmowa	xvii
Wprowadzenie	xix
Rozdział 1. Podstawowe pojęcia	1
Terminy związane z pamięcią	3
Alokacja statyczna	10
Maszyna rejestrowa	11
Stos	12
Maszyna stosowa	19
Wskaźnik	23
Serta	25
Ręczne zarządzanie pamięcią	28
Automatyczne zarządzanie pamięcią	34
Alokator, mutator i kolektor	36
Zliczanie referencji	42
Kolektor śledzący	48
Faza oznaczania (Mark)	49
Faza zbierania (Collect)	54
Nieco historii	58
Podsumowanie	61
Zasada 1 – Ucz się	61
Rozdział 2. Niskopoziomowe zarządzanie pamięcią	63
Sprzęt	64
Pamięć	71
Procesor (CPU)	73
System operacyjny	94
Pamięć wirtualna	94
Duże strony	99

Fragmentacja pamięci wirtualnej	100
Ogólny układ pamięci	101
Zarządzanie pamięcią w Windows	103
Układ pamięci systemu Windows	109
Zarządzanie pamięcią w systemie Linux	112
Układ pamięci w systemie Linux	114
Wpływ systemu operacyjnego	116
NUMA i grupy procesorów	117
Podsumowanie	119
Zasada 2 – Należy unikać losowego dostępu, a promować dostęp sekwencyjny	119
Zasada 3 – Poprawiaj przestrzenną i czasową lokalność danych	120
Zasada 4 – Używaj bardziej zaawansowanych możliwości	121
Rozdział 3. Pomiary pamięci	123
Mierzyć wcześniej	125
Obciążenie i inwazyjność	126
Próbkowanie i śledzenie	126
Drzewo wywołań	127
Grafy obiektów	128
Statystyki	130
Opóźnienie a przepustowość	134
Zrzuty pamięci, śledzenie, debugowanie na żywo	136
Środowisko systemu Windows	137
Przegląd	137
VMMap	138
Liczniki wydajności	139
Śledzenie zdarzeń dla systemu Windows	146
Zestaw narzędzi Windows Performance Toolkit	160
PerfView	173
ProcDump, DebugDiag	184
WinDbg	185
Deasemblerzy i dekompileatory	188
BenchmarkDotNet	189
Narzędzia komercyjne	191
Środowisko Linux	202
Przegląd	202
Perfcollect	204
Trace Compass	206
Zrzuty pamięci	217
Podsumowanie	219
Zasada 5 – Mierz odświeżanie pamięci wcześniej	221

Rozdział 4. Podstawy .NET	223
Wersje .NET	224
Wewnętrzne mechanizmy .NET	227
Dogłębna analiza przykładowego programu	231
Podzespoły i domeny aplikacji	238
Usuwalne podzespoły	240
Regiony pamięci procesu	241
Scenariusz 4-1. Jak dużo pamięci zajmuje mój program?	247
Scenariusz 4-2. Zużycie pamięci przez nasz program stale rośnie	249
Scenariusz 4-3. Zużycie pamięci przez nasz program stale rośnie	252
Scenariusz 4-4. Zużycie pamięci przez nasz program stale rośnie	255
System typów	259
Kategorie typów	259
Przechowywanie typów	262
Typy wartościowe	263
Typy referencyjne	273
Łańcuchy znaków	281
Internowanie łańcuchów znaków	288
Scenariusz 4-5. Zużycie pamięci przez nasz program jest zbyt duże	295
Opakowywanie i rozpakowywanie	298
Przekazywanie przez referencję	304
Przekazywanie wystąpienia typu wartościowego przez referencję	304
Przekazywanie wystąpienia typu referencyjnego przez referencję	306
Lokalność danych dla typów	307
Dane statyczne	311
Pola statyczne	311
Wewnętrzne działanie danych statycznych	312
Podsumowanie	318
Struktury	318
Klasy	319
Rozdział 5. Partycjonowanie pamięci	323
Strategie partycjonowania	324
Partycjonowanie według rozmiaru	326
Serta małych obiektów	328
Serta dużych obiektów	328
Partycjonowanie ze względu na czas życia	333
Scenariusz 5-1. Czy mój program jest zdrowy? Rozmiary generacji w czasie	339
Pamiętane zbiory	344
Tabele kart	350
Pakiety kart	357

Partycjonowanie fizyczne	360
Scenariusz 5-2. Wyciek pamięci w nopCommerce?	367
Scenariusz 5-3. Marnowanie miejsca w stercie dużych obiektów?	378
Anatomia segmentów i sterty	380
Ponowne wykorzystanie segmentów	383
Podsumowanie	387
Zasada 11 – Monitoruj rozmiary generacji	387
Zasada 12 – Unikaj niepotrzebnych referencji na sterzie	388
Zasada 13 – Monitoruj użycie segmentów	389
Rozdział 6. Alokacja pamięci	391
Wprowadzenie do alokacji	392
Alokacja z przesuwaniem wskaźnika	393
Alokacja oparta na liście wolnych obszarów	401
Tworzenie nowego obiektu	406
Alokacja na sterzie małych obiektów	408
Alokacja na sterzie dużych obiektów	412
Równoważenie stert	416
Wyjątek OutOfMemoryException	420
Scenariusz 6-1. Brak pamięci	421
Alokacje na stosie	424
Unikanie alokacji	426
Jawne alokacje typów referencyjnych	428
Ukryte alokacje	458
Różne ukryte alokacje wewnątrz bibliotek	468
Scenariusz 6-2. Badanie alokacji	474
Scenariusz 6-3. Azure Functions	478
Podsumowanie	479
Zasada 14 – Unikaj alokacji na sterzie na ścieżkach kodu krytycznych ze względu na wydajność	480
Zasada 15 – Unikaj nadmiernych alokacji na sterzie LOH	480
Zasada 16 – Preferuj alokacje na stosie, gdy jest to właściwe	481
Rozdział 7. Wprowadzenie do odświeżania pamięci	483
Widok wysokopoziomowy	484
Proces odświeżania pamięci na przykładzie	485
Kroki procesu odświeżania pamięci	493
Scenariusz 7-1. Analizowanie wykorzystania odświeżania pamięci	494
Profilowanie odświeżania pamięci	499
Dane dostrajające wydajność odświeżania pamięci	501
Dane statyczne	501
Dane dynamiczne	504
Scenariusz 7-2. Zrozumienie budżetu alokacji	508

Wyzwalacze odśmiecania	519
Wyzwalanie przez alokację	520
Wyzwalanie jawne	521
Scenariusz 7-3. Analizowanie jawnych wywołań odśmiecania pamięci	526
Wyzwalacz systemowy przy niskim poziomie pamięci	533
Różne wyzwalacze wewnętrzne	534
Wstrzymywanie silnika wykonawczego	535
Scenariusz 7-4. Analizowanie czasów wstrzymania	538
Generacja do skazania	540
Scenariusz 7-5. Analiza skazanych generacji	544
Podsumowanie	546
 Rozdział 8. Odśmiecanie pamięci – faza oznaczania	547
Przechodzenie przez graf obiektów i oznaczanie	547
Korzenie w zmiennych lokalnych	549
Przechowywanie zmiennych lokalnych	550
Korzenie na stosie	551
Zakres leksykalny	552
Żywe korzenie na stosie a zakres leksykalny	553
Żywe korzenie na stosie z gorliwym odśmiecaniem korzeni (eager root collection)	555
Informacje GC	563
Przypięte zmienne lokalne	569
Skanowanie korzeni na stosie	572
Korzenie finalizacji	573
Wewnętrzne korzenie GC	574
Korzenie uchwytów GC	575
Obsługa wycieków pamięci	584
Scenariusz 8-1. Wyciek pamięci w nopCommerce?	586
Scenariusz 8-2. Identyfikowanie najpopularniejszych korzeni	590
Podsumowanie	593
 Rozdział 9. Odśmiecanie pamięci – faza planowania	595
Sterta małych obiektów	596
Zasłepki (plugs) i luki (gaps)	596
Scenariusz 9-1. Zrzut pamięci z nieprawidłowymi strukturami	602
Tabela klocków (brick table)	604
Przypinanie	606
Scenariusz 9-2. Badanie przypinania	612
Granice generacji	618
Degradowanie	619
Sterta dużych obiektów	624
Zasłepki i luki	624

Decydowanie o kompaktowaniu	627
Podsumowanie	629
Rozdział 10. Odsmięcanie pamięci – zmiatanie i kompaktowanie	631
Faza zmiatania	631
Sterta małych obiektów	632
Sterta dużych obiektów	633
Faza kompaktowania	634
Sterta małych obiektów	634
Sterta dużych obiektów	640
Scenariusz 10-1 Fragmentacja sterty dużych obiektów	640
Podsumowanie	651
Zasada 17 – Obserwuj wstrzymania środowiska uruchomieniowego	652
Zasada 18 – Unikaj „kryzysu wieku średniego”	653
Zasada 19 – Unikaj fragmentacji w starej generacji i na sterzie LOH	654
Zasada 20 – Unikaj jawnego odsmięcania pamięci	655
Zasada 21 – Unikaj wycieków pamięci	655
Zasada 22 – Unikaj przypinania	656
Rozdział 11. Odmiany odsmięcania pamięci	659
Przegląd trybów	659
Tryb stacji roboczej a tryb serwera	659
Tryb niewspółbieżny a współbieżny	662
Konfiguracja trybów	663
Platforma .NET Framework	664
Platforma .NET Core	665
Przerwa na odsmięcanie pamięci i obciążenie procesora	667
Opisy trybów	670
Niewspółbieżny tryb stacji roboczej (Workstation Non-Concurrent)	670
Współbieżny tryb stacji roboczej (Workstation Concurrent, przed wersją 4.0)	672
Tryb stacji roboczej w tle (Background Workstation)	674
Niewspółbieżny tryb serwera (Server Non-Concurrent)	685
Tryb serwera w tle (Background Server)	687
Tryby opóźnień (latency modes)	689
Tryb wsadowy	690
Tryb interakcyjny	690
Niskie opóźnienie	691
Trwałe niskie opóźnienie	692
Region bez odsmięcania pamięci	694
Cele optymalizacji opóźnień	697
Wybieranie odmiany odsmięcania pamięci	698
Scenariusz 11-1. Sprawdzanie ustawień odsmięcania pamięci	700

Scenariusz 11-2. Testowanie różnych trybów odświeżania pamięci	703
Zasada 23 – Świadomie wybieraj tryb odświeżania pamięci	712
Zasada 24 – Pamiętaj o trybach opóźnień	713
Rozdział 12. Czas życia obiektów	715
Cykl życia obiektów i zasobów	716
Finalizacja	718
Wprowadzenie	718
Problem z gorliwym odświeżaniem korzeni	724
Finalizatory krytyczne	728
Wewnętrzne działanie finalizacji	729
Scenariusz 12-1. Wyciek pamięci przy finalizacji	739
Wskrzyszanie	747
Obiekty wykorzystujące wzorzec Disposable	751
Bezpieczne uchwytty (safe handles)	759
Słabe referencje	766
Buforowanie (caching)	772
Wzorzec słabych zdarzeń	775
Scenariusz 12-2. Wyciek pamięci z powodu zdarzeń	783
Podsumowanie	786
Zasada 25 – Unikaj finalizatorów	787
Zasada 26 – Preferuj jawne czyszczenie	788
Rozdział 13. Różnorodne zagadnienia	791
Uchwytty zależne	792
Pamięć lokalna wątku	799
Pola statyczne wątku	800
Sloty na dane wątku	804
Wewnętrzne działanie pamięci lokalnej wątku	805
Scenariusze użycia	814
Wskaźniki zarządzane	815
Zmienne lokalne ref	817
Zwracane wartości ref	818
Zmienne ref tylko do odczytu i parametry in	820
Wewnętrzne działanie typów ref	826
Wskaźniki zarządzane w C# – zmienne ref	841
Więcej na temat struktur	849
Struktury tylko do odczytu	850
Struktury ref (typy w rodzaju byref)	852
Bufory o stałych rozmiarach	855
Układ obiektów/struktur	860

Ograniczenie unmanaged	873
Typy kopiowalne (blittable)	879
Podsumowanie	882
Rozdział 14. Techniki zaawansowane	883
Span<T> i Memory<T>	883
Span<T>	884
Memory<T>	903
IMemoryOwner<T>	907
Wewnętrzne działanie Memory<T>	913
Wskazówki dotyczące Span<T> i Memory<T>	916
Typ Unsafe	916
Wewnętrzne działanie Unsafe	923
Projektowanie zorientowane na dane	924
Projektowanie taktyczne	926
Projektowanie strategiczne	930
Więcej na temat przyszłości	943
Typy referencyjne dopuszczające wartość null	943
Potoki	951
Podsumowanie	959
Rozdział 15. Programowe interfejsy API	961
Interfejs API dla odśmiecania pamięci	962
Dane i statystyki dotyczące odśmiecania pamięci	962
Powiadomienia ze strony odśmiecania pamięci	973
Kontrolowanie obciążenia pamięci niezarządzanej	976
Jawne odśmiecanie	976
Regiony bez odśmiecania pamięci	977
Zarządzanie finalizacją	977
Użycie pamięci	977
Wewnętrzne wywołania w klasie GC	979
Hosting środowiska uruchomieniowego CLR	980
ClrMD	992
Biblioteka TraceEvent	1000
Niestandardowe odśmiecanie pamięci	1003
Podsumowanie	1008
Indeks	1011