

Spis treści

Podziękowania	7
Wstęp	9
Krótką historia powstawania wielowątkowości	10
Wyjaśnienie pojęć związanych z wątkami	11
Rozdział 1. Programowanie wielowątkowe	13
1.1. Klasa Thread	15
1.1.1. Wykonywanie pracy w tle	16
1.2. Klasa ThreadPool	17
1.2.1. Rodzaje kolejek	17
1.2.2. Metoda UnsafeQueueUserWorkItem	18
1.3. Klasa Task	19
1.3.1. Porównanie z klasą Thread	19
1.3.2. Porównanie z klasą ThreadPool	20
1.3.3. Metody Wait, WaitAll i WaitAny	20
1.3.4. Właściwość Result	20
1.3.5. Metoda ContinueWith	21
1.3.6. Opcja AttachedToParent	21
1.3.7. Metoda StartNew właściwości Factory	22
1.3.8. Metoda Run	22
1.3.9. Enumeracja TaskStatus	24

1.4. Klasa TaskFactory	26
1.5. Struktura CancellationToken	27
1.5.1. Korzystanie ze struktury CancellationToken	27
1.6. Klasa CancellationTokenSource	29
1.7. Klasa Timer	30
1.8. Klasa TaskCompletionSource	32
1.9. Klasa SynchronizationContext	33
1.10. Klasa TaskScheduler	35
ĆWICZENIA DO ROZDZIAŁU 1.	37
Rozdział 2. Programowanie równoległe	39
2.1. Klasa Parallel	41
2.1.1. Metoda For	41
2.1.2. Metoda ForEach	41
2.1.3. Metoda Invoke	42
2.2. Technologia PLINQ	43
2.3. Klasa Partitioner	45
2.3.1. Optymalizacja krótkich operacji	46
2.4. Porównanie z klasą Task	47
ĆWICZENIA DO ROZDZIAŁU 2.	48
Rozdział 3. Programowanie asynchroniczne	49
3.1. Transformacja kodu asynchronicznego	51
3.2. Słowo kluczowe await	52
3.2.1. Porównanie z metodą ContinueWith	52
3.2.2. Użycie wraz z metodą Run	53
3.2.3. Współbieżność await	54
3.3. Słowo kluczowe async	55
3.3.1. Asynchroniczne wyrażenie lambda	55
3.3.2. Metoda z sygnaturą async void	56
3.3.3. Opis wykonywania się metody asynchronicznej	56
3.3.4. Sposoby radzenia sobie z wielokrotnymi wywołaniami	57
3.3.5. Sztuczna synchroniczność i asynchroniczność	58
3.4. Asynchroniczność wewnątrz LINQ	59
3.5. Zadania zakończone	60

3.6. Metoda Yield	61
3.6.1. Porównanie z właściwością CompletedTask	61
3.7. Interfejsy asynchroniczne	62
3.7.1. Interfejs IAsyncEnumerable<T>	62
3.7.2. Interfejs IAsyncDisposable	63
3.8. Własna implementacja	64
3.9. Rady dotyczące programowania asynchronicznego	65
3.9.1. Używanie metody ConfigureAwait	65
3.9.2. Wykonywanie metody asynchronicznej synchronicznie	65
3.9.3. Użycie await bezpośrednio przed zwróceniem metody	66
3.9.4. Asynchroniczność w konstruktorze	66
3.9.5. Przeciążenie przyjmujące delegat Func<Task>	67
3.9.6. Bardzo długo wykonująca się praca	67
3.10. Struktura ValueTask	68
3.10.1. Interfejs IValueTaskSource	68
3.10.2. Konsumowanie ValueTask	69
3.10.3. Porównanie z klasą Task	70
ĆWICZENIA DO ROZDZIAŁU 3.	71
Rozdział 4. Synchronizacja	73
4.1. Podstawowe elementy synchronizacji	75
4.1.1. Klasa Volatile	75
4.1.2. Klasa Interlocked	76
4.2. Blokady trybu jądra	80
4.3. Blokady hybrydowe	81
4.3.1. Przekazywanie instancji do metod klasy Monitor	81
4.3.2. Słowo kluczowe lock	81
4.4. Blokady asynchroniczne	83
4.5. Leniwa inicjalizacja	84
4.5.1. Blokada z podwójnym sprawdzeniem	84
4.5.2. Klasa Lazy<T>	85
4.5.3. Klasa ThreadLocal<T>	85
4.6. Kolekcje współbieżne	87
4.6.1. Klasa BlockingCollection<T>	87
ĆWICZENIA DO ROZDZIAŁU 4.	90

ROZWIĄZANIA	91
ROZWIĄZANIA DO ROZDZIAŁU 1.	91
ROZWIĄZANIA DO ROZDZIAŁU 2.	94
ROZWIĄZANIA DO ROZDZIAŁU 3.	95
ROZWIĄZANIA DO ROZDZIAŁU 4.	97
Źródła	101