

Opis

Współbieżność jest jednym z wymogów nowoczesnych aplikacji, niezależnie od ich rodzaju i platformy. Tworzenie współbieżnych aplikacji jest co prawda dość złożone, jednak cecha ta czyni interfejs użytkownika bardziej responsywnym, a systemy — skalowalnymi. Zrozumienie istoty współbieżności i umiejętność jej zaimplementowania jest już koniecznością dla profesjonalnego dewelopera, nawet jeśli zagadnienia te wydają się trudne i skomplikowane.

Ta książka przedstawia nowoczesne podejście do współbieżności. Jest przeznaczona dla programistów, którzy chcą poszerzyć swoją wiedzę i umiejętności. Omówiono tu zagadnienia wielowątkowości i przedstawiono kilka różnych rodzajów współbieżności, w tym programowanie równoległe, asynchroniczne i reaktywne. Opisano biblioteki, dzięki którym programowanie współbieżnych aplikacji staje się znacznie łatwiejsze poprzez podniesienie poziomu abstrakcji. Zawarte w książce receptury uzupełniono o działający kod i przedyskutowano sposób ich działania.

W książce między innymi:

- Omówienie różnych rodzajów współbieżności
- Reaktywność i przepływ zdarzeń
- Biblioteka zadań równoległych (TPL)
- Biblioteki rozszerzeń reaktywnych w LINQ
- Prowadzenie testów jednostkowych za pomocą kodu współbieżnego
- Asynchroniczne programowanie obiektowe
- Synchronizacja wątków

Kod współbieżny — responsywność, skalowalność, nowoczesność!

Stephen Cleary jest doświadczonym programistą. Od samego początku tworzy oprogramowanie open source, między innymi bibliotek Boost C++. Opublikował również kilka własnych bibliotek i narzędzi.

Spis treści

- Opinie o książce Współbieżność w języku C#. Receptury
- Przedmowa
 - Kto powinien przeczytać tę książkę?
 - Dlaczego napisałem tę książkę?
 - Poruszanie się po tej książce
 - Źródła online
 - Konwencje użyte w tej książce
 - Podziękowania
- Rozdział 1. Współbieżność: przegląd

- 1.1. Wprowadzenie do współbieżności
- 1.2. Wprowadzenie do programowania asynchronicznego
- 1.3. Wprowadzenie do programowania równoległego
- 1.4. Wprowadzenie do programowania reaktywnego (Rx)
- 1.5. Wprowadzenie do przepływów danych
- 1.6. Wprowadzenie do programowania wielowątkowego
- 1.7. Kolekcje dla aplikacji współbieżnych
- 1.8. Nowoczesne projektowanie
- 1.9. Podsumowanie informacji na temat kluczowych technologii
- Rozdział 2. Podstawy async
 - 2.1. Wstrzymanie na określony czas
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.2. Zwracanie wykonanych zadań
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.3. Raportowanie postępu
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.4. Oczekiwanie na wykonanie zestawu zadań
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.5. Oczekiwanie na wykonanie jakiegokolwiek zadania
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.6. Przetwarzanie wykonanych zadań
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.7. Unikanie kontekstu dla kontynuacji
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 2.8. Obsługa wyjątków z metod async typu Task
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również

- 2.9. Obsługa wyjątków z metod async typu void
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 3. Podstawy przetwarzania równoległego
 - 3.1. Równoległe przetwarzanie danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 3.2. Równoległa agregacja
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 3.3. Równoległe wywołanie
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 3.4. Równoległość dynamiczna
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 3.5. Parallel LINQ
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 4. Podstawy przepływu danych
 - 4.1. Łączenie bloków
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 4.2. Propagowanie błędów
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 4.3. Usuwanie połączeń między blokami
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 4.4. Ograniczanie pojemności bloków
 - Problem
 - Rozwiązanie

- Dyskusja
 - Zobacz również
 - 4.5. Przetwarzanie równoległe za pomocą bloków przepływu danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 4.6. Tworzenie niestandardowych bloków
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 5. Podstawy Rx
 - 5.1. Konwersja zdarzeń .NET
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 5.2. Wysyłanie powiadomień do kontekstu
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 5.3. Grupowanie danych zdarzeń za pomocą okienek i buforów
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 5.4. Ujarzmianie strumieni zdarzeń za pomocą ograniczania przepływu i próbkowania
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 5.5. Limity czasu
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 6. Testowanie
 - 6.1. Testy jednostkowe metod async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 6.2. Testy jednostkowe metod async, które powinny zakończyć się niepowodzeniem
 - Problem
 - Rozwiązanie
 - Dyskusja

- Zobacz również
 - 6.3. Testy jednostkowe metod async void
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 6.4. Testy jednostkowe siatek przepływu danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 6.5. Testy jednostkowe strumieni obserwowalnych Rx
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 6.6. Testy jednostkowe strumieni obserwowalnych Rx za pomocą atrapy harmonogramu
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 7. Interoperacyjność
 - 7.1. Metody opakowujące async dla metod async ze zdarzeniami Completed
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 7.2. Metody opakowujące async dla metod Begin/End
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 7.3. Metody opakowujące async dla dowolnych operacji lub zdarzeń
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 7.4. Metody opakowujące async dla kodu równoległego
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 7.5. Metody opakowujące async dla strumieni obserwowalnych Rx
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 7.6. Metody opakowujące strumieni obserwowalnych Rx dla kodu asynchronicznego

- Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- 7.7. Strumienie obserwowalne Rx i siatki przepływu danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 8. Kolekcje
 - 8.1. Niemutowalne stosy i kolejki
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.2. Listy niemutowalne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.3. Zbiory niemutowalne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.4. Słowniki niemutowalne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.5. Słowniki bezpieczne wątkowo
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.6. Kolejki blokujące
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.7. Stosy i multizbiory blokujące
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 8.8. Kolejki asynchroniczne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również

- 8.9. Stosy i multizbiory asynchroniczne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- 8.10. Kolejki blokujące/asynchroniczne
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 9. Anulowanie
 - 9.1. Wysyłanie żądań anulowania
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.2. Reagowanie na żądania anulowania poprzez odpytywanie
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.3. Anulowanie z powodu przekroczenia limitu czasu
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.4. Anulowanie kodu async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.5. Anulowanie kodu równoległego
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.6. Anulowanie kodu reaktywnego
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.7. Anulowanie siatek przepływu danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 9.8. Wstrzykiwanie żądań anulowania
 - Problem
 - Rozwiązanie
 - Dyskusja

- Zobacz również
 - 9.9. Współdziałanie z innymi systemami anulowania
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 10. Przyjazne funkcyjnie programowanie obiektowe
 - 10.1. Interfejsy async i dziedziczenie
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 10.2. Konstruowanie async: fabryki
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 10.3. Konstruowanie async: wzorzec inicjowania asynchronicznego
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 10.4. Właściwości async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 10.5. Zdarzenia async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 10.6. Usuwanie async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 11. Synchronizacja
 - 11.1. Blokady
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 11.2. Blokady async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 11.3. Sygnały blokujące
 - Problem

- Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 11.4. Sygnały async
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 11.5. Ograniczanie współbieżności
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 12. Planowanie
 - 12.1. Planowanie pracy dla puli wątków
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 12.2. Wykonywanie kodu za pomocą dyspozytora zadań
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 12.3. Planowanie kodu równoległego
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 12.4. Synchronizacja przepływu danych z wykorzystaniem dyspozytorów
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- Rozdział 13. Scenariusze
 - 13.1. Inicjowanie współdzielonych zasobów
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 13.2. Odroczone ewaluacja Rx
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
 - 13.3. Asynchroniczne wiązanie danych
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również

- 13.4. Stan niejawny
 - Problem
 - Rozwiązanie
 - Dyskusja
 - Zobacz również
- O autorze
- Kolofon