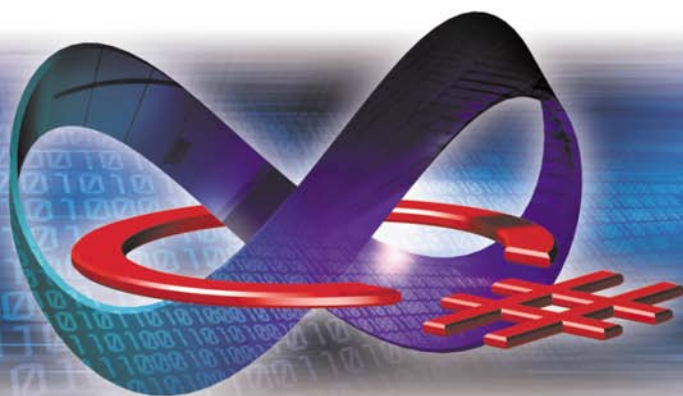


Visual Studio 2010

dla programistów C#



Środowisko **Visual Studio 2010**, język **C# 4.0**
i podstawy projektowania aplikacji

Bazy danych i technologia **LINQ**

Nowy paradygmat zrównoleglenia aplikacji z biblioteką **TPL**

Technologie **Windows** i wprowadzenie do **XNA 4.0**

Zawiera CD



Jacek Matulewski, Dawid Borycki, Grzegorz Krause,
Maciej Grabek, Maciej Pakulski, Mateusz Warczak,
Jacek Lewandowski, Sławomir Orłowski

» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

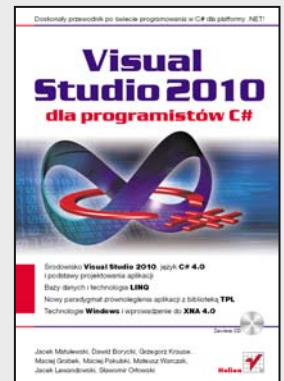
- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Visual Studio 2010 dla programistów C#

Autor: [Jacek Matulewski](#)
ISBN: 978-83-246-2173-6
Format: 172×245, stron: 824



Doskonały przewodnik po świecie programowania w C# dla platformy .NET!

- Środowisko Visual Studio 2010, język C# 4.0 i podstawy projektowania aplikacji
- Bazy danych i technologia LINQ
- Nowy paradygmat zrównoleglenia aplikacji z biblioteką TPL
- Technologie Windows i wprowadzenie do XNA 4.0

Język C# na dobre zadomowił się już na rynku programistycznym. Zarówno on sam, jak i platforma .NET przydają się do najróżniejszych zastosowań, obejmujących nie tylko projektowanie złożonych aplikacji korzystających z baz danych, ale również usług systemu Windows lub gier. Pora więc zapoznać się z tym językiem programowania, oswoić ze środowiskiem Visual Studio 2010 firmy Microsoft i zorientować się, jak możesz wykorzystać nowe umiejętności. Ta obszerna książka o przekrojowym charakterze ułatwi Ci wejście w świat języka C#, platform .NET i XNA. Poznasz również wiele zaawansowanych technik programistycznych.

Z pierwszej części przewodnika poznasz język C# i bogate środowisko programistyczne Visual Studio 2010. Nauczysz się również jak wydajnie projektować aplikacje z graficznym interfejsem użytkownika. Kolejne części poruszają kwestie połączenia aplikacji z bazą danych i stosowania technologii LINQ. W dobie komputerów z wieloma procesorami o wielu rdzeniach bardzo ważna jest umiejętność tworzenia aplikacji wielowątkowych. Temu zagadnieniu poświęcona jest część czwarta dotycząca nowej biblioteki TPL. Znajdziesz tu także opis zagadnień związanych z usługami sieciowymi WCF i pracą z wykorzystaniem różnorodnych technologii Windows. Osobną część poświęcono także wprowadzeniu do technologii XNA 4.0 i budowaniu z jej pomocą gier korzystających z grafiki 2D. Jeśli chcesz programować w języku C#, nie możesz przegapić tej książki!

- Środowisko Visual Studio 2010, język C# i debugowanie kodu
- Projektowanie zorientowane obiektowo i przegląd komponentów Windows Forms
- Podstawy ADO.NET, eksport i prezentacja danych, transakcje i raportowanie
- Wprowadzenie do zapytań LINQ i tworzenie źródeł danych LINQ
- Programowanie współbieżne – wątki i zadania
- Klasa Parallel i dane w programach równoległych, technologia PLINQ
- Biblioteka WCF – tworzenie usług sieciowych nowej generacji
- Technologie Windows – rejestr systemu, komunikaty, usługi i automatyzacja
- Tworzenie gier w XNA 4.0, użycie shakerów HLSL
- Użycie wyrażeń regularnych – sposób na weryfikację wprowadzanych danych
- Testy jednostkowe – najlepsza metoda sprawdzania poprawności kodu

Poznaj platformę .NET i język C# od podszewki!

Spis treści

Wstęp	15
Część I Projektowanie aplikacji Windows	17
Rozdział 1. Środowisko Visual Studio 2010	19
Projektowanie interfejsu aplikacji	20
Tworzenie projektu	20
Dokowanie palety komponentów Toolbox	22
Tworzenie interfejsu za pomocą komponentów Windows Forms	22
Zapisywanie i wczytywanie projektu	24
Analiza kodu pierwszej aplikacji	24
Metody zdarzeniowe	29
Metoda uruchamiana w przypadku wystąpienia zdarzenia kontrolki	29
Testowanie metody zdarzeniowej	29
Przypisywanie istniejącej metody do zdarzeń komponentów	31
Edycja metody zdarzeniowej	32
Modyfikowanie własności komponentów	32
Wywoływanie metody zdarzeniowej z poziomu kodu	32
Reakcja aplikacji na naciskanie klawiszy	33
Rozdział 2. Debugowanie kodu	35
Skąd biorą się błędy i jak ich unikać?	35
Kontrolowane uruchamianie aplikacji w Visual C#	36
Śledzenie wykonywania programu krok po kroku (F10 i F11)	37
Run to Cursor (Ctrl+F10)	38
Breakpoint (F9)	38
Okna Locals i Watch	39
Stan wyjątkowy	41
Zgłaszanie wyjątków	41
Przechwytywanie wyjątków w konstrukcji try..catch	42
Rozdział 3. Język C#	45
Platforma .NET	46
Środowisko uruchomieniowe	46
Kod pośredni i podwójna kompilacja	46
Skróty, które warto poznać	46
Podstawowe typy danych	47
Deklaracja i zmiana wartości zmiennej	47
Typy liczbowe oraz znakowy	48
Określanie typu zmiennej przy inicjacji (pseudotyp var)	49
Operatory	49
Konwersje typów podstawowych	51

Operatory is i as	51
Łańcuchy	52
Typ wyliczeniowy	54
Leniwe inicjowanie zmiennych	55
Metody	55
Przeciążanie metod	56
Domyślne wartości argumentów metod — argumenty opcjonalne (nowość języka C# 4.0)	57
Argumenty nazwane (nowość języka C# 4.0)	58
Wartości zwracane przez metody	58
Zwracanie wartości przez argument metody	58
Delegacje i zdarzenia	59
Wyrażenia lambda	60
Typy wartościowe i referencyjne	61
Nullable	62
Pudełkowanie	63
Typy dynamiczne (nowość języka C# 4.0)	63
Sterowanie przepływem	66
Instrukcja warunkowa if..else	66
Instrukcja wyboru switch	66
Pętle	67
Wyjątki	68
Dyrektywy preprocesora	70
Kompilacja warunkowa — ostrzeżenia	70
Definiowanie stałych preprocesora	70
Błoki	71
Atrybuty	71
Kolekcje	72
„Zwykłe” tablice	72
Pętla foreach	74
Sortowanie	74
Kolekcja List	75
Kolekcja SortedList i inne słowniki	77
Kolejka i stos	77
Tablice jako argumenty metod oraz metody z nieokreśloną liczbą argumentów	78
Słowo kluczowe yield	79
Nowa forma inicjacji obiektów i tablic	80
Rozdział 4. Projektowanie zorientowane obiektowo	83
Przykład struktury (Ułamek)	84
Przygotowanie projektu	84
Konstruktor i statyczne obiekty składowe	84
Pierwsze testy	85
Konwersje na łańcuch (metoda ToString) i na typ double	86
Metoda upraszczająca ułamek	86
Własności	87
Operatory arytmetyczne	88
Operatory porównania oraz metody Equals i GetHashCode	89
Operatory konwersji	90
Implementacja interfejsu (na przykładzie IComparable)	91
Definiowanie typów parametrycznych	92
Definiowanie typów ogólnych	93
Określanie warunków, jakie mają spełniać parametry	94
Implementacja interfejsów przez typ ogólny	95
Definiowanie aliasów	96
Typy ogólne z wieloma parametrami	97
Rozszerzenia	98
Typy anonimowe	99

Rozdział 5. Przegląd komponentów biblioteki Windows Forms	101
Notatnik.NET	101
Projektowanie interfejsu aplikacji i menu główne	101
Okna dialogowe i pliki tekstowe	106
Edycja i korzystanie ze schowka	113
Drukowanie	113
Elektroniczna kukułka	120
Ekran powitalny (splash screen)	120
Przygotowanie ikony w obszarze powiadomiania	122
Odtwarzanie pliku dźwiękowego	125
Ustawienia aplikacji	126
Dywan graficzny	129
Lista uruchomionych procesów	132
Rozdział 6. Przeciągnij i upuść	135
Podstawy	135
Interfejs przykładowej aplikacji	135
Inicjacja procesu przeciągania	137
Akceptacja upuszczenia elementu	138
Reakcja na upuszczenie elementu	139
Czynności wykonywane po zakończeniu procesu przenoszenia i upuszczania	140
Przenoszenie elementów między różnymi aplikacjami	140
Zagadnienia zaawansowane	140
Opóźnione inicjowanie procesu przenoszenia	141
Przenoszenie wielu elementów	142
Przenoszenie plików	144
Rozdział 7. Przezroczyste okna o dowolnym kształcie	147
Konfiguracja formy	147
Wczytywanie obrazu	148
Przezroczystość i łagodne znikanie okna	150
Zamykanie klawiszem Esc	151
Przenoszenie formy za dowolny punkt	151
Menu kontekstowe	152
Rozdział 8. Projektowanie kontrolki	155
Komponent FileListBox	156
Implementacja podstawowych funkcjonalności	156
Rozbudowa komponentu o możliwość zmiany katalogu	163
Właściwości	164
Zdarzenia — interakcja z komponentem	168
Odświeżanie komponentu i automatyczne śledzenie zmian w prezentowanym katalogu	173
Kompilacja komponentu do postaci biblioteki DLL	175
Prosty przykład wykorzystania komponentu FileListBox: przeglądanie plików tekstowych	179
Kolorowy pasek postępu	181
Tworzenie projektu	182
Rysowanie obramowania kontrolki	182
Pola i własności	182
Rysowanie paska postępu	185
Metody	186
Zdarzenia	186
Rozdział 9. Studium przypadku: implementacja liczb zespolonych i ich użycie do rysowania fraktali	189
Implementacja liczb zespolonych	189
Projekt struktury	190
Własności	192

Operatory	193
Metody statyczne	195
Testy	197
Rysowanie zbiorów Mandelbrota i Julii	198
Trochę teorii	198
Implementacja	200

Część II Technologie bazodanowe ADO.NET 203

Rozdział 10. Podstawy ADO.NET 205

Podstawy relacyjnych baz danych	205
Technologia ADO.NET	206
Instalacja bazy Northwind	207
Połączenie ze źródłem danych	207
Modyfikacja danych w obiekcie DataSet	211
Tworzenie nowych rekordów	211
Edycja rekordów	215
Wersjonowanie obiektu DataRow i kontrola wprowadzanych danych	216
Usuwanie rekordów	220
Obiekt TableAdapter jako pomost między DataSet a źródłem danych	221
Parametryzacja kwerend	221
Edycja danych przy użyciu obiektu TableAdapter	224
Integralność danych i ich usuwanie	224
Wstawianie danych z wykorzystaniem obiektu TableAdapter	228

Rozdział 11. Prezentacja danych 233

Mechanizm DataBinding — wiązanie danych z kontrolkami	233
Mechanizm DataBinding a komponent Chart	237
Komponent DataGridView	239
Formatowanie danych wyświetlanych w komponencie DataGridView	240
Tworzenie formularza z podformularzem (Master/Detail Form)	246

Rozdział 12. Eksport danych 249

Informacje wstępne	249
Eksport danych przy użyciu schowka systemowego	250
Wstawianie danych do arkusza kalkulacyjnego z pominięciem schowka systemowego	253
Transfer danych poprzez plik XML	254
Eksport danych do formatu HTML	256

Rozdział 13. Transakcje 257

Tworzenie i ręczna kontrola transakcji	257
Automatyczne zarządzanie transakcjami	261
Konkurencyjność i poziom izolacji transakcji	264

Rozdział 14. Bezpieczeństwo w aplikacjach bazodanowych 265

Szyfrowanie połączenia	265
Zabezpieczenie kodu pośredniego przy użyciu Code Access Security (CAS)	270
ConnectionString a sposoby uwierzytelnienia w serwerze Microsoft SQL Server	274
Uwagi końcowe dotyczące projektowania aplikacji bazodanowych	275

Rozdział 15. Raportowanie 277

Pierwszy raport	277
Konfiguracja źródła danych	278
Projektowanie raportu	279
Kolumny obliczeniowe raportu	280
Stosowanie formatu prezentacji danych	281
Grupowanie danych	282

Projektowanie graficznego układu raportu	283
Etykiety, rysunki i listy	284
Prezentacja danych w postaci macierzowej	287
Parametry raportu i filtrowanie danych	289
Filtr definiowany na poziomie raportu	289
Filtrowanie raportu na etapie pobierania danych źródłowych	290
Kluczowe wskaźniki efektywności	292
Wykresy	294
Tworzenie raportu zawierającego podraport	295
Eksport danych przy użyciu raportu	297

Część III LINQ 303

Rozdział 16. Wprowadzenie do zapytań LINQ na przykładzie kolekcji (LINQ to Objects) 305

Pobieranie danych (filtrowanie i sortowanie)	306
Najprostsza prezentacja pobranych danych	307
Analiza pobranych danych	307
Wybór elementu	307
Weryfikowanie danych	308
Prezentacja w grupach	308
Łączenie zbiorów danych	308
Łączenie danych z różnych źródeł w zapytaniu LINQ — operator join	309
Możliwość modyfikacji danych źródła	309

Rozdział 17. LINQ to DataSet 311

Konfiguracja kontrolki DataSet	312
LINQ to DataSet, czyli tam i z powrotem	313
Rozszerzenie AsEnumerable klasy DataTable	316
Obliczenia wykonywane na danych z tabeli	316
Dowolność sortowania i filtrowania pobieranych danych	316

Rozdział 18. LINQ to SQL 317

Klasa encji	317
Pobieranie danych	319
Aktualizacja danych w bazie	319
Modyfikacje istniejących rekordów	320
Dodawanie i usuwanie rekordów	320
Inne operacje	321
Wizualne projektowanie klasy encji	322
O/R Designer	322
Współpraca z kontrolkami tworzącymi interfejs aplikacji	325
Kreator źródła danych i automatyczne tworzenie interfejsu użytkownika	325
Łączenie danych z dwóch tabel — operator join	328
Relacje (Associations)	328
Korzystanie z procedur składowanych	331
Pobieranie danych za pomocą procedur składowanych	331
Modyfikowanie danych za pomocą procedur składowanych	331
Wykonywanie dowolnych poleceń SQL	332

Rozdział 19. Trzy sposoby na odczytywanie i zapisywanie danych w plikach XML 335

Podstawy języka XML	335
Deklaracja	335
Elementy	336
Atrybuty	336
Komentarze	336

Klasy XmlTextReader i XmlTextWriter	336
Zapis do pliku XML	337
Odczyt danych z pliku XML	338
Analiza i odczyt pliku XML o nieznannej strukturze	339
Serializacja obiektów do pliku XML	341
Serializacja obiektu do pliku XML	342
Deserializacja obiektu z pliku XML	343
XML i ADO.NET	343
Wczytywanie danych z pliku XML do komponentu DataSet	344
Zapisywanie zmian do pliku XML za pośrednictwem DataSet	345
LINQ to XML	346
Tworzenie pliku XML za pomocą klas XDocument i XElement	346
Pobieranie wartości z elementów o znanej pozycji w drzewie	347
Przenoszenie danych z kolekcji do pliku XML	349
Przenoszenie danych z bazy danych (komponentu DataSet) do pliku XML	350
Zapytania LINQ	350
Modyfikacja pliku XML	351
Rozdział 20. Tworzenie źródeł danych LINQ	353
Źródło liczb losowych	353
IEnumerable	353
IEnumerable<>	355
Oddzielenie źródła od jego interfejsu	357
IQueryable i IQueryable	359
IQueryable<> i IQueryable<>	360
Drzewo wyrażenia	361
Tabela w pliku tekstowym, czyli LINQ to TXT	367
Proste rozwiązanie	367
Plan projektu „pełnego” źródła danych	370
Klasa odpowiedzialna za odczytanie pliku tekstowego	370
Modelowanie danych (mapowanie typów)	374
Projekt źródła danych i analizator zapytania	377
Przetwarzanie danych z analizatora	384
Edycja danych w źródle LINQ i dodawanie nowych rekordów	393
Przykład wykorzystania biblioteki LINQ to TXT	397
Rozdział 21. Entity Framework	401
Podstawy Entity Framework	401
Entity Framework — pierwszy projekt	402
Plik .edmx	405
Praca z obiektami Entity Framework. LINQ to Entities and Entity SQL	409
Pierwsze zapytanie oparte na modelu encji	409
Filtrowanie danych	411
Projekcja danych	412
Grupowanie danych	414
Związki między encjami w zapytaniach	415
Sortowanie wyników zapytania	417
Wykorzystanie programowania równoległego w zapytaniach LINQ to Entities	418
Tworzenie, modyfikowanie i usuwanie obiektów encji	419
Inne funkcje Entity Framework	421
Zachłanne i leniwe ładowanie (Eager and lazy loading)	421
Wykorzystanie procedur składowanych do modyfikacji encji	424
Wykorzystanie procedur składowanych do pobierania danych	428
Tworzenie tabel w bazie danych na podstawie modelu encji	431
Dziedziczenie typu tabela na hierarchię	434
Dziedziczenie typu tabela na typ	438
Entity Framework i POCO	441

Część IV Programowanie współbieżne	447
Rozdział 22. Wątki	449
Monte Carlo	449
Obliczenia bez użycia dodatkowych wątków	450
Przeniesienie obliczeń do osobnego wątku	451
Usypianie wątku	452
Przerywanie działania wątku (Abort)	453
Wstrzymywanie i wznowianie działania wątku	454
Wątki działające w tle	455
Zmiana priorytetu wątku	455
Użycie wielu wątków i problemy z generatorem liczb pseudolosowych	456
Czekanie na ukończenie pracy wątku (Join)	458
Sekcje krytyczne (lock)	460
Przesyłanie danych do wątku	461
Pula wątków	463
Jeszcze raz o komunikacji między wątkami	465
Synchronizacja wątków — rejestr braków	466
Korzystanie z muteksów w celu zapobiegania uruchamianiu wielu instancji aplikacji	466
Rozdział 23. Zadania — nowy przepis na programowanie współbieżne w platformie .NET 4.0	469
Tworzenie zadania	469
Praca z zadaniami	470
Dane przekazywane do zadań	471
Dane zwracane przez zadania	472
Przykład: test liczby pierwszej	472
Synchronizacja zadań	473
Przykład: sztafeta zadań	474
Przerywanie zadań	475
Stan zadania	478
Fabryka zadań	480
Planista i zarządzanie kolejkowaniem zadań	482
Ustawienia zadań	485
Rozdział 24. Klasa Parallel. Zrównoleglanie pętli	487
Równoległa pętla for	487
Równoległa pętla foreach	489
Metoda Invoke	489
Ustawienia pętli równoległych. Klasa ParallelOptions	490
Przerywanie pętli za pomocą CancellationToken	490
Kontrola wykonywania pętli	491
Synchronizacja pętli równoległych. Obliczanie π metodą Monte Carlo	492
Wielowątkowa klasa Random	496
Rozdział 25. Dane w programach równoległych	499
Praca ze zbiorami danych w programowaniu równoległym	499
Współbieżne struktury danych	499
Kolekcja CollectionBag	500
Współbieżne kolejka i stos	501
Praca z BlockingCollection	501
Własna kolekcja współbieżna	503
Agregacja	506
Agregacje dla kolekcji równoległych	507
PLINQ — zrównoleglone zapytania LINQ	511
Przykład zapytania PLINQ	511
Jak działa równoległe LINQ?	512

Kiedy PLINQ jest wydajne?	514
Metody przekształcające dane wynikowe	515
Przerywanie zapytań	515
Metoda ForAll	517

Część V Usługi sieciowe, czyli WCF od A do C 519

Potrzeba matką wynalazku	520
--------------------------------	-----

Rozdział 26. WCF — jak to ugryźć? 523

Podstawy działania	523
Biblioteki i przestrzenie nazw	523
WCF = A + B + C	524
C jak contract	524
B jak binding	524
A jak address	528
Pierwszy serwis	529
Hosting	534
Self-Hosting	534
IIS	538
Serwis windowsowy	540
WAS	542
Klient	542

Rozdział 27. Narzędzia i konfiguracja 547

Narzędzia	547
Konfiguracja kluczem do wszystkiego — bliższe spojrzenie	550
<behaviors>	551
<bindings>	551
<client>	553
<commonBehaviors>	553
<extensions>	553
<routing>	553
<serviceHostingEnvironment>	554
<service>	554
<standardEndpoints>	554

Rozdział 28. Wybrane nowości WCF w platformie .NET 4.0 557

Uproszczona konfiguracja	557
Discovery	558
Serwis routujący	562

Część VI Technologie Windows 569

Rozdział 29. Rejestr systemu Windows 571

Korzystanie z rejestru	571
Odczytywanie danych z rejestru	571
Zapisywanie oraz odczytywanie położenia i rozmiaru formy w prywatnym kluczu aplikacji ..	573
Usuwanie klucza z rejestru	575
Przeglądarka skojarzeń plików	575
Informacja o typach plików przechowywana w rejestrze	575
Przygotowanie interfejsu	577
Odczytywanie listy rozszerzeń	578
Pobieranie opisu, polecenia głównego i domyślnego edytora dla podanego typu plików	579

Rozdział 30. Zarządzane biblioteki DLL i mechanizm odzwierciedlenia 583

Tworzenie zarządzanej biblioteki DLL	584
Projekt biblioteki DLL	584
Dodawanie referencji do biblioteki systemowej platformy .NET	584

Wyświetlanie informacji o systemie i platformie .NET	585
Statyczne ładowanie bibliotek DLL	586
Dołączanie do projektu bibliotek DLL użytkownika	586
Dynamiczne ładowanie zarządzanych bibliotek DLL i dynamiczne rozpoznawanie typów	587
Dynamiczne ładowanie zarządzanej biblioteki .dll	587
Analiza zawartości biblioteki załadowanej dynamicznie	588
Weryfikacja obecności w bibliotece DLL klasy o znanej nazwie	589
Lista metod w klasie z biblioteki DLL	589
Weryfikacja obecności konkretnej metody w klasie z biblioteki DLL	590
Lista argumentów wybranej metody	591
Uruchamianie metody statycznej z klasy wczytanej z biblioteki DLL	593
Uruchamianie metody na rzecz instancji obiektu. Przekazywanie parametrów i odczytywanie zwracanej wartości	593
Użycie typu dynamic	595
Korzystanie z bibliotek DLL jako wtyczek (MEF)	595
Gospodarz i wtyczka w jednej aplikacji	595
Rozdzielenie gospodarza i wtyczki	598
Katalog	599
Wiele wtyczek implementujących ten sam interfejs	600
Metadane	601
Rozdział 31. Mechanizm PInvoke	603
Funkcja bez argumentów	603
Problemy z argumentami	605
Zwracanie wartości przez argumenty	606
Zwracanie tablicy znaków w funkcjach WinAPI	608
Rozdział 32. Komunikaty Windows	609
Wysyłanie komunikatów Windows	609
Identyfikacja aplikacji	609
Wysyłanie komunikatu do okna o znanym uchwycie	611
Komunikaty jako sposób porozumiewania się z systemem	611
Odbieranie komunikatów Windows	612
Monitor komunikatów	612
Reakcja na wybrany komunikat	613
Rozdział 33. Integracja kodu zarządzanego i niezarządzanego	615
Kod zarządzany (nadzorowany)	616
Import funkcji z biblioteki systemowej. Wersja bez zabezpieczeń	617
Import funkcji z biblioteki systemowej. Wersja z zabezpieczeniem	619
Import wielu funkcji z biblioteki DLL. Klasa opakowująca	621
Import wielu funkcji z biblioteki DLL. Ładowanie bibliotek DLL	625
Różne wersje funkcji WinAPI	627
Konwersja typów zarządzanych i niezarządzanych	628
Przekazywanie struktur do funkcji niezarządzanych	628
Projekt własnej biblioteki DLL. Integracja projektu zarządzanego i niezarządzanego	632
Przekazywanie ciągów znakowych i wskaźników bez użycia struktury IntPtr	634
Wywołanie zwrotne i kopiowanie obszarów pamięci kodu niezarządzanego	635
Rozdział 34. Usługi Windows	637
Tworzenie usługi	638
Instalacja usługi	641
Przygotowanie projektu i ręczna instalacja usługi	641
Projekt instalatora usługi	644
Odczytywanie błędów generowanych w trakcie pracy usługi	645
Zarządzanie usługą z poziomu innej aplikacji	646

Rozdział 35. Automatyzacja	649
Technologie COM	649
Excel jako serwer automatyzacji	650
Pobranie informacji o aktualnie uruchomionej aplikacji Excel	651
Uruchamianie i zamykanie serwera automatyzacji MS Excel	652
Eksplorowanie danych w arkuszu kalkulacyjnym	654
Korzystanie z okien dialogowych serwera automatyzacji. Zapisywanie danych w pliku	655
Zapisywanie danych z wykorzystaniem okna dialogowego aplikacji klienckiej	656
Edycja danych w komórkach Excela	656
Obsługa zdarzeń serwera automatyzacji	658
Korzystanie z funkcji matematycznych i statystycznych Excela	659
Serwer automatyzacji Microsoft Word	661
Uruchamianie aplikacji Microsoft Word i tworzenie nowego dokumentu lub otwieranie istniejącego	661
Wywoływanie funkcji Worda na przykładzie sprawdzania pisowni i drukowania	662
Wstawianie tekstu do bieżącego dokumentu Worda	663
Zapisywanie bieżącego dokumentu Worda	663
Zaznaczanie i kopiowanie całego tekstu dokumentu Worda do schowka	664
Kopiowanie zawartości dokumentu Worda do komponentu RichTextBox bez użycia schowka (z pominięciem formatowania tekstu)	664
Formatowanie zaznaczonego fragmentu tekstu w dokumencie Worda	665
Serwer automatyzacji przeglądarki Internet Explorer	665
Projektowanie własnego serwera automatyzacji	667
Przykład prostego projektu serwera automatyzacji	667
Testy	669
Rozdział 36. Odtwarzanie multimediów przy użyciu technologii ActiveX	671
Odtwarzanie plików multimedialnych	671
Wstrzymywanie, wznowianie i prezentowanie informacji o postępie odtwarzania pliku	672
Wybór odtwarzanego pliku	674
Kontrola głośności odtwarzania	675
Przewijanie odtwarzanego pliku	675
Widok pełnego ekranu	676
Implementacja listy odtwarzanych plików	676
Rozdział 37. Visual Studio Tools for Office	681
Elementy języka C# 4.0 ułatwiające tworzenie dodatków dla pakietu Office	682
Szablon projektu dodatku do skoroszytu	682
Komponenty Windows Forms w arkuszu Excela	684
Tworzenie wykresu w skoroszycie przy użyciu dodatku napisanego w C#	686
Modyfikacja wstążki menu w aplikacji MS Excel	687
Ukrycie standardowych zakładki menu w aplikacji Excel	690
Tworzenie panelu zadań aplikacji MS Excel	690
Formatowanie komórek	693
Obsługa zdarzeń	695
Dodatki na poziomie aplikacji	696
Interfejs aplikacji	696
Implementacja funkcjonalności aplikacji	698
Zarządzanie dodatkami w aplikacji MS Excel	702
Instalacja dodatku	702
Usunięcie dodatku	702
Podsumowanie	703

Część VII Grafika 2D w XNA 4.0	705
Rozdział 38. Struktura projektu gry w XNA	707
Pola klasy gry	708
Metody gry	708
Pętla gry	709
Rozdział 39. Wyświetlanie obrazów	713
Rysowanie obrazu za pomocą obiektu SpriteBatch. Klasa Sprite	714
Układ współrzędnych	719
Rozdział 40. Wykrywanie kolizji	721
Animacja piłki	721
Wykrywanie i obsługa kolizji	723
Rozdział 41. Obsługa kontrolerów gry (klawiatura, mysz i gamepad)	727
Rozdział 42. Komponenty gry	731
Rozdział 43. Oprawa dźwiękowa	739
XACT	739
MediaPlayer	745
Rozdział 44. Wyświetlanie tekstu	747
Rozdział 45. Gra z komputerem	751
Rozdział 46. Shadery i efekty w XNA	755
Proces renderowania	755
Podstawy języka HLSL	756
Typy zmiennych	756
Semantyka	757
Sterowanie przepływem	758
Definiowanie funkcji	758
Efekty, techniki, przebiegi	758
Przykład efektu HLSL i jego wykorzystanie w grze XNA	759
Kilka przykładów efektów	762
Drżenie ekranu w Pongu	764
Dodatek A Aplikacje konsolowe i informacje o systemie	767
Klasa Console	767
Projekt aplikacji konsolowej	767
Drukowanie napisów w konsoli	768
Czekanie na akceptację użytkownika	769
Odczytywanie danych z klawiatury	769
Komunikat „okienkowy” w aplikacji konsolowej	770
Informacje o środowisku aplikacji	771
Podstawowe informacje o systemie i profilu użytkownika	771
Katalogi specjalne zdefiniowane w bieżącym profilu użytkownika	772
Odczytywanie zmiennych środowiskowych	773
Lista dysków logicznych	773
Dodatek B Instalacja SQL Server 2008 Express Edition i bazy danych AdventureWorks	775
Instalacja SQL Server 2008 Express Edition	775
Instalacja bazy danych AdventureWorks	779
Uruchamianie SQL Server Management Studio	779
Dodatek C Wyrażenia regularne	781
Podstawowe elementy konstrukcji wyrażeń regularnych	781
Odnajdywanie znaków specjalnych	781
Odnajdywanie znaków i cyfr	782

Odnajdywanie znaków w wyznaczonych miejscach łańcucha	782
Określanie ilości wystąpień podanych znaków	783
Przykłady tworzenia wzorców	783
Aplikacja	783
Kod pocztowy	784
Imię żeńskie	785
Walidacja adresu URL	786
Odszukiwanie powtarzających się wyrazów	787
Modyfikacja łańcucha znakowego	787
Dodatek D Testy jednostkowe	789
Praktyka przygotowywania testów na przykładzie struktury Complex	790
Metody inicjujące i czyszczące	794
Obiekty pozorne	794
Skorowidz	795

Rozdział 12.

Eksport danych

Dawid Borycki

Informacje wstępne

Często spotykanym elementem aplikacji bazodanowych jest możliwość eksportowania danych do pliku obsługiwane przez zewnętrzną aplikację, np. w postaci arkusza kalkulacyjnego lub dokumentu przenośnego (PDF). Najpopularniejszym programem do tworzenia arkuszy kalkulacyjnych jest Microsoft Excel, element pakietu Microsoft Office. Naszą uwagę skupimy na rozszerzeniu możliwości aplikacji przygotowanej w poprzednim rozdziale o możliwość eksportu danych do pliku w formacie Excel i plików stron WWW. Eksport do plików PDF omówimy w rozdziale 15., dotyczącym tworzenia raportów.

Transfer danych do plików **.xls* lub **.xlsx* można zrealizować na kilka różnych sposobów. Jednak najbardziej elastycznym sposobem jest wykorzystanie mechanizmu automatyzacji, który oprócz transferu danych umożliwi również formatowanie wyeksportowanych danych. Automatyzacja pozwala na wysłanie danych do Excela na kilka różnych sposobów. Oto one.

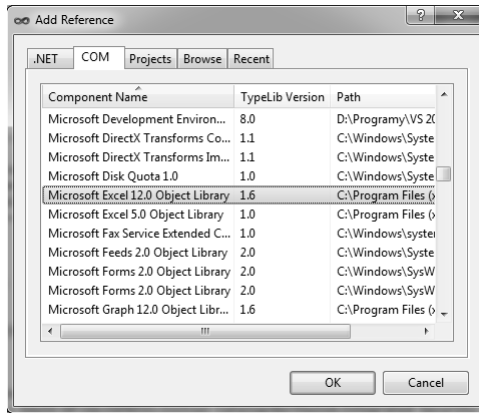
1. Wykorzystanie schowka systemowego. Jest to najprostszy sposób. Jednak czasochłonny eksport dużej ilości danych może doprowadzić do sytuacji, w której dane znajdujące się w schowku mogą zostać zmodyfikowane przez inną aplikację. Jeśli bowiem użytkownik komputera korzysta jednocześnie z edytora tekstu i w momencie transferu danych do Excela umieści w schowku jakieś inne dane, zostaną one skopiowane do arkusza kalkulacyjnego. Scenariusz ten można porównać do działania aplikacji wielowątkowej, w której kilka wątków (Word, Excel) uzyskuje jednocześnie dostęp do współdzielonego zasobu (schowka). Brak synchronizacji wątków i ograniczenia dostępu do wspólnego zasobu może doprowadzić do niepożądanych skutków.
2. Przeniesienie pojedynczych komórek, które polega na wykorzystaniu obiektu `Range`, reprezentującego zakres komórek w arkuszu kalkulacyjnym. Ten sposób umożliwia wstawienie danych w dowolnym miejscu w arkuszu. Wykorzystanie tego podejścia jest zalecane przy eksporcie małych ilości danych, gdyż każdorazowe użycie obiektu `Range` skutkuje wysłaniem żądania do odpowiadającego mu interfejsu, co spowalnia transfer danych.
3. Eksport macierzy danych. Przenoszenie pojedynczych komórek można znacznie przyspieszyć przez wykorzystanie metody `Range.get_Resize`, której zadaniem jest zaznaczenie docelowego zakresu komórek, w którym umieszczone zostaną eksportowane dane. Dzięki tej metodzie zmniejsza się ilość żądań do interfejsu `Range`.
4. Transfer danych za pomocą pliku tekstowego. W tym przypadku komórki tabeli są zapisywane do pliku tekstowego, a następnie importowane do arkusza kalkulacyjnego. Takie podejście wydaje się być najbardziej korzystne dla architektury klient-serwer w przypadku transferu dużej ilości informacji. Dane zostają wygenerowane po stronie serwera, a importowane przez klienta.

5. Eksport danych przy użyciu pliku XML (ang. *Extensible Markup Language*). Projektanci aplikacji Excel, począwszy od wersji 2002, udostępnili metody, które umożliwiają bezpośrednie otwarcie pliku XML. Gdy dane umieszczone są w obiekcie DataSet, ich eksport do pliku XML sprowadza się do użycia metody WriteXML. Tak wygenerowany plik może zostać otwarty w aplikacji Excel ręcznie lub z poziomu kodu.

W poniższych podrozdziałach zawarto przykłady implementacji możliwości nr 1, 3 i 5, gdyż to właśnie one wydają się najbardziej efektywne. Wszystkie poniższe projekty wymagają referencji do biblioteki zawierającej typ aplikacji Excel, w której umieszczone są informacje dotyczące obiektów, metod i interfejsów udostępnianych przez tę aplikację. W celu dodania referencji klikamy pozycję *Add reference* w menu *Project*, a następnie na zakładce COM odszukujemy wpis o nazwie *Microsoft Excel NN.N Object Library*, gdzie NN.N oznacza wersję zainstalowanego w systemie Excela (rysunek 12.1).

Rysunek 12.1.

Dodanie referencji do Excela



Eksport danych przy użyciu schowka systemowego

W tym punkcie wykorzystamy schowek systemowy jako medium transmisji danych na potrzeby eksportu informacji z komponentu DataGridView do arkusza kalkulacyjnego. W tym celu należy wykonać następujące czynności:

1. Uzupełnij interfejs użytkownika aplikacji rozwijanej w poprzednim rozdziale o przycisk oraz komponent OpenFileDialog.
2. Własność Filter kontrolki OpenFileDialog zmień na:
Arkusze kalkulacyjne[*.*xls;*.xlsx]
We własności FileName umieść wpis *Arkusz1*.
3. W pliku *Form1.cs* umieść polecenie importujące przestrzeń nazw związaną z aplikacją *Excel*:
`using Excel = Microsoft.Office.Interop.Excel;`
4. Klasę *Form1* uzupełnij o następujące pola:
`Excel.Application xlApp;`
5. Otwarcie pliku arkusza kalkulacyjnego, do którego eksportujesz dane, realizuje metoda z listingu 12.1; jej definicję umieść w klasie *Form1*.

Listing 12.1. Otwarcie pliku docelowego

```
bool otworzPlikExcela(out Excel.Workbook wb, string sciezka = null)
{
    // Jeśli Excel został wcześniej uruchomiony,
    // to 'podłączamy' się do uruchomionej instancji.
    // W przeciwnym wypadku tworzymy nową instancję.
    try
    {
        xlApp = (Excel.Application)System.Runtime.InteropServices
```



```

        .Marshal.GetActiveObject("Excel.Application");
    }
    catch (Exception ex)
    {
        xlApp = new Microsoft.Office.Interop.Excel.Application();
    }

    wb = null;

    if (sciezka != null)
    {
        try
        {
            wb = xlApp.Workbooks.Open(sciezka);
        }
        catch (Exception ex)
        {
            try
            {
                wb = (Excel.Workbook)System.Runtime.InteropServices.
                    Marshal.BindToMoniker(sciezka);
            }
            catch (Exception ex1)
            {
                return false;
            }
        }
    }

    return true;
}

```

6. Domyślną metodę zdarzeniową przycisku zdefiniuj według wzoru przedstawionego na listingu 12.2.

Listing 12.2. Transfer danych do arkusza kalkulacyjnego przy użyciu schowka

```

private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
    {
        MessageBox.Show("Wybierz poprawną ścieżkę do eksportu danych");
        return;
    }
    else if (!openFileDialog1.CheckFileExists)
    {
        MessageBox.Show("Podany plik " + openFileDialog1.FileName
            + " nie istnieje");
        return;
    }

    Excel.Workbook wb = null;
    if (!otworzPlikExcela(openFileDialog1.FileName, out wb))
    {
        MessageBox.Show("Próba otwarcia pliku:\n" +
            openFileDialog1.FileName +
            "\nNie powiodła się", "Formularz");
    }

    // Przygotowanie eksportowanych danych
    string daneDowstawienia = "";
    foreach (DataGridViewRow wiersz in dataGridView1.Rows)
    {
        foreach (DataGridViewCell komorka in wiersz.Cells)
            daneDowstawienia += komorka.Value + "\t";

        daneDowstawienia += '\n';
    }
}

```

```

// Kopiowanie danych do schowka
Clipboard.SetDataObject(daneDoWstawienia);

// Pobranie pierwszego arkusza
Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets.get_Item(1);

// Wklejanie zawartości schowka rozpoczynamy od komórki A1
Excel.Range adres = ws.get_Range("A1");
ws.Paste(adres, daneDoWstawienia);

// Zapis pliku i zakończenie połączenia z Excelem
wb.Save();
xlApp.Quit();
System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);

MessageBox.Show("Pomyślnie wyeksportowano dane do pliku:\n"
    + openFileDialog1.FileName,
    "Formularz");
}

```

Warto zwrócić uwagę na kilka elementów powyższego rozwiązania, a przede wszystkim na metodę `otwórzPlikExcela`, której definicję przedstawiłem na listingu 12.1. W pierwszym kroku metoda ta dokonuje sprawdzenia, czy aplikacja Excel została już uruchomiona; wykorzystuje do tego celu funkcję `GetActiveObject` obiektu `Marshal`. Ten ostatni dostarcza szereg metod umożliwiających przeprowadzanie różnego rodzaju operacji między obiektami kodu zarządzanego i niezarządzanego. Kodem zarządzanym jest kod uruchamiany pod kontrolą środowiska uruchomieniowego CLR (ang. *Common Language Runtime*), czyli kod pośredni (ang. *Microsoft Intermediate Language — MSIL*), do którego kompilowane są wszystkie aplikacje pisane dla platformy .NET. W przypadku mechanizmu automatyzacji napotykamy bowiem na konieczność wywołania z poziomu kodu zarządzanego metod eksponowanych przez niezarządzone komponenty COM. W takich sytuacjach niezbędne staje się alokowanie i kopiowanie niezarządzanej pamięci, konwersja typów niezarządzanych do zarządzanych (i *vice versa*) oraz odwoływanie się do szerokiego spektrum metod udostępnianych przez obiekty niezarządzone.

Wróćmy jednak do meritum sprawy. Przedstawię teraz zasadę działania metody `GetActiveObject`. Funkcja ta przeszukuje systemową tablicę ROT (ang. *Running Object Table*) w celu odnalezienia obiektu podanego jako argument funkcji `GetActiveObject`. Zadaniem tej tablicy jest śledzenie wszystkich uruchomionych obiektów, dzięki czemu zamiast uruchamiać kolejne instancje aplikacji, możemy „podłączyć się” do już uruchomionej instancji wybranego obiektu.

Funkcja `otwórzPlikExcela` tworzy nowy obiekt typu `Excel.Application`, gdy na podstawie danych zawartych w tablicy ROT stwierdzono brak uruchomionej instancji Excela. Funkcja `GetActiveObject` zgłasza wówczas wyjątek. Z tego powodu wywołanie tej funkcji zostało opatrzone blokiem `try, catch`, gdzie w części `catch` dokonujemy utworzenia nowego obiektu `Excel.Application`, zamiast wyświetlenia komunikatu o błędzie.

W kolejnym kroku dokonujemy próby otwarcia pliku wskazanego w argumentcie `nazwaPliku`. Służy do tego metoda `Workbooks.Open` obiektu `Excel.Application`. Jeśli w trakcie próby otwarcia pliku okaże się, że został on wcześniej otwarty, wykonanie metody nie powiedzie się. Przechwytyując zgłoszony wyjątek, funkcja `otwórzPlikExcela` próbuje „podłączyć się” do wskazanego pliku; wykorzystuje do tego celu metodę `BindToMoniker` obiektu `Marshal`. Ta ostatnia zwraca referencję do interfejsu identyfikowanego przez jej parametr o nazwie `MonikerName` (*przezwisko*) określający typ obiektu, do którego chcemy się podłączyć. Skoroszyt programu Excel zawiera wiele obiektów. Dlatego też w celu poinformowania metody `BindToMoniker`, który z elementów składowych chcemy wykorzystać, należy użyć rzutowania.

W tym momencie warto zwrócić uwagę na pewien nowy aspekt języka C# w wersji 4.0, wykorzystany w metodzie `otwórzPlikExcela`. Chodzi o omówione już w rozdziale 3. parametry opcjonalne, dobrze znane programistom C++, które ułatwiają przeciążanie metod. Przyczyną wykorzystania tych parametrów jest fakt, że niektóre funkcje tylko nieznacznie różnią się między sobą. Więc zamiast powielać kod, pisząc kolejne funkcje, lepiej wykorzystać istniejącą procedurę, której działanie można zmodyfikować za pomocą domyślnych wartości nadanych parametrom opcjonalnym. Przykładem może być funkcja `otwórzPlikExcela`;

w funkcji tej parametrem opcjonalnym jest argument *ścieżka*, którego wartością domyślną jest `null` (listing 12.1). Taka konstrukcja funkcji pozwala na jej późniejsze wykorzystanie w celu podłączenia się do aplikacji Excel, bez konieczności przekazywania argumentu *ścieżka*, co zostanie zastosowane w następnym projekcie.

Gdyby nie parametry opcjonalne (a z taką sytuacją spotykamy się w poprzednich wersjach języka C#), wywołanie funkcji `Workbooks.Open` wymagałoby przekazania (w miejscu parametrów opcjonalnych) specjalnego typu `System.Type.Missing`, tj.:

```
object missing = System.Type.Missing;
xlApp.Workbooks.Open(ścieżka, missing, missing, missing, missing, missing, missing,
    missing, missing, missing, missing, missing, missing, missing);
```

Wróćmy teraz do technologii ADO.NET i komponentu `DataGridView`, który zawiera dane do eksportu. W celu ich wyłuskania funkcja z listingu 12.2 wykonuje iterację po wszystkich komórkach obiektu `DataGridView`, a zawarte w nich informacje kopiuje do łańcucha znakowego. Obiekt ten, za pomocą funkcji `Clipboard.SetDataObject`, zostaje wstawiony do schowka systemowego, po to by za chwilę został wklejony do docelowego arkusza kalkulacyjnego.

W przykładzie przedstawionym na listingach 12.1 i 12.2 eksportowane dane nie zawierają wiersza nagłówka. Można go jednak łatwo utworzyć przy użyciu następujących poleceń:

```
string wierszNaglowka = "";
for (int i = 0; i < dataGridView1.ColumnCount; i++)
    wierszNaglowka += dataGridView1.Columns[i].Name + "\t";
wierszNaglowka += "\n";
```

Po czym w parametrze metody `Clipboard.SetDataObject` należy umieścić obiekt będący konkatenacją łańcuchów `wierszNaglowka` i `daneDoWstawienia`, a mianowicie:

```
Clipboard.SetDataObject(wierszNaglowka + daneDoWstawienia);
```

Warunkiem poprawnego wykonania się kodu przedstawionego na listingach 12.1 i 12.2 jest wcześniejsze utworzenie pliku z arkuszem kalkulacyjnym, do którego eksportujemy dane. W następnym projekcie dane zawarte w komponentcie `DataGridView` zostaną wyeksportowane do nowego pliku.

Wstawianie danych do arkusza kalkulacyjnego z pominięciem schowka systemowego

Wykorzystanie schowka systemowego do eksportu danych powoduje niedogodności, o których pisałem we wstępie do tego rozdziału. Jednak w Excelu udostępniono obiekt typu `Range`, który umożliwia bezpośrednią modyfikację zawartości odpowiednich komórek. Do tego celu wystarczy zmienić jego własność `Value` na zadaną wartość. Prześledzimy to w kolejnym przykładzie:

1. Na formularzu aplikacji umieść kolejny przycisk.
2. W klasie `Form1` utwórz pole typu `string`, przechowujące nazwę docelowego arkusza kalkulacyjnego, tj.:


```
string nazwaPliku = "dane.xlsx";
```
3. Na podstawie metody `otworzPlikExcela`, napisanej w poprzednim podrozdziale, w klasie `Form1` umieść metodę `podlaczDoExcela`, zdefiniowaną w listingu 12.3.

Listing 12.3. „Podłączenie” do uruchomionej instancji aplikacji Excel

```
bool podlaczDoExcela()
{
    Excel.Workbook wb = null;
    return otworzPlikExcela(out wb);
}
```

4. W ostatnim kroku utwórz domyślną metodę zdarzeniową do przycisku i umieść w niej polecenia z listingu 12.4.

Listing 12.4. Eksport danych z komponentu *DataGridView* do nowego pliku w formacie *Excela*

```

private void button2_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
    {
        MessageBox.Show("Wybierz poprawną ścieżkę do eksportu danych");
        return;
    }

    if (!podlaczDoExcela())
    {
        MessageBox.Show("Próba eksportu danych nie powiodła się" , "Formularz");
        return;
    }

    xlApp.Workbooks.Add();
    Excel.Workbook wb = xlApp.Workbooks.get_Item(1);
    Excel.Worksheet ws = wb.Worksheets.get_Item(1);

    // Tablica zawierająca dane do przeniesienia, w której
    // dodatkowy wiersz przeznaczono na nagłówek tabeli
    object[,] daneDoWstawienia =
        new Object[dataGridView1.RowCount + 1, dataGridView1.ColumnCount];

    for(int i = 0; i < dataGridView1.RowCount; i++)
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            {
                if(i == 0)
                    daneDoWstawienia[i, j] = dataGridView1.Columns[j].Name;
                else
                    daneDoWstawienia[i, j] = dataGridView1.Rows[i - 1].Cells[j].Value;
            }

    Excel.Range adres = ws.get_Range("A1");

    // Dopasowanie docelowego zakresu komórek do rozmiarów tablicy daneDoWstawienia
    adres = adres.get_Resize(dataGridView1.RowCount + 1, dataGridView1.ColumnCount);
    adres.Value = daneDoWstawienia;

    string sciezka = folderBrowserDialog1.SelectedPath;

    if (sciezka.EndsWith("\\"))
        sciezka += nazwaPliku;
    else
        sciezka += "\\" + nazwaPliku;

    wb.SaveAs(sciezka);
    wb.Close();
    xlApp.Quit();
    System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);
    MessageBox.Show("Pomyślnie wyeksportowano dane do pliku:\n"
        + sciezka, "Formularz");
}

```

Transfer danych poprzez plik XML

Język XML pozwala zapisać do pliku tekstowego dane w hierarchicznej strukturze. Pliki sformatowane przy jego użyciu są łatwe do zinterpretowania i przenaszalne na różne systemy, dlatego język ten cieszy się dużym zainteresowaniem programistów. Z tego powodu również i Excel¹ udostępnia możliwość odczytu i zapisu plików zapisanych w tym formacie. Wykorzystamy to w tym rozdziale do eksportu danych z obiektu *DataSet*. W tym celu interfejs użytkownika projektowanej aplikacji uzupełniamy o kolejny przycisk. Klikając nań dwukrotnie, utworzymy jego domyślną metodę zdarzeniową, w której umieścimy polecenia z listingu 12.5.

¹ Od wersji 2002.

Listing 12.5. *Transfer danych do Excela przy użyciu pliku XML*

```

private void button3_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() != System.Windows.Forms.DialogResult.OK)
    {
        MessageBox.Show("Wybierz poprawną ścieżkę do eksportu danych");
        return;
    }

    if (!podlaczDoExcela())
    {
        MessageBox.Show("Próba eksportu danych nie powiodła się". "Formularz");
        return;
    }

    // Tworzenie tymczasowego pliku XML
    string plikTymczasowy = "temp.xml";
    System.IO.FileStream fs =
        new System.IO.FileStream(plikTymczasowy, System.IO.FileMode.Create);

    // Zapis pierwszej tabeli do tymczasowego pliku XML
    dataSet.Tables[0].WriteXml(fs);
    fs.Close();

    // Wyłączenie wyświetlania komunikatów Excela i otwarcie pliku XML
    object missing = System.Type.Missing;
    xlApp.DisplayAlerts = false;

    Excel.Workbook wb =
        xlApp.Workbooks.OpenXML(System.IO.Path.GetFullPath(plikTymczasowy),
            missing, Excel.XlXmlLoadOption.xlXmlLoadImportToList);

    // Zapis pliku w formacie Excela
    string sciezka = folderBrowserDialog1.SelectedPath;

    if(sciezka.EndsWith("\\"))
        sciezka += nazwaPliku;
    else
        sciezka += "\\ " + nazwaPliku;

    wb.SaveAs(sciezka);
    wb.Close();
    xlApp.Quit();
    System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);

    // Usunięcie pliku tymczasowego
    System.IO.File.Delete(plikTymczasowy);

    MessageBox.Show("Pomyślnie wyeksportowano dane do pliku: " +
        sciezka, "Formularz");
}

```

Działanie powyższej funkcji opiera się na wykorzystaniu metody `WriteXML` obiektu `DataSet`, umożliwiającej utworzenie pliku XML na podstawie danych, zawartych w lokalnym repozytorium reprezentowanym przez `DataSet`. Funkcja ta została użyta do zapisu pierwszej tabeli w obiekcie `DataSet`, tj. tabeli `Customers` w postaci pliku XML.

Kolejnym istotnym zagadnieniem wymagającym wyjaśnienia jest funkcja `Workbooks.OpenXML`. Jej pierwszy parametr określa ścieżkę otwieranego pliku XML, drugi jest pomijany w trakcie automatyzacji Excela, można go używać tylko z poziomym makr.

Ostatni i najbardziej istotny argument określa sposób otwarcia pliku XML. Dostępne możliwe sposoby otwarcia pliku XML zawarto w typie wyliczeniowym `XlXmlLoadOption`, w którym znajdują się następujące stałe:

1. `xlXmlLoadImportToList` — wskazuje, by Excel zaimportował zawartość pliku XML jako tabelę danych. Ten sposób wydaje się najkorzystniejszy w przypadku eksportu danych z aplikacji bazodanowej.
2. `xlXmlLoadMapXml` — wyświetla schemat danych zawartych w pliku XML na pasku zadań programu Excel. Użytkownik może następnie otworzyć plik Excela i zdecydować, które kolumny tabeli danych mają być widoczne w arkuszu kalkulacyjnym.
3. `xlXmlLoadOpenXml` — otwiera plik XML, korzystając ze specjalnego algorytmu wyrównywania danych (ang. *flattening algorithm*). Excel dostosowuje w ten sposób zawartość pliku XML do struktury dwuwymiarowego arkusza kalkulacyjnego, złożonego z kolumn i wierszy. Wyrównywanie to jest wymagane, gdyż struktura plików XML może mieć więcej niż dwa wymiary.
4. `xlXmlLoadPromptUser` — pozwala użytkownikowi wybrać jedną z powyższych opcji.

W naszym przykładzie korzystamy z pierwszej opcji, ponieważ tabela danych ma strukturę dwuwymiarową, identyczną ze strukturą arkusza kalkulacyjnego. Dane zawarte w pliku XML wygenerowanym przy użyciu metody `DataSet.WriteXML` Excel może poprawnie zaimportować do arkusza kalkulacyjnego.

W celu zablokowania komunikatów wyświetlanych przez aplikację Excel własność `DisplayAlerts` ustawiamy na `false`. Pozwala to na pozbycie się komunikatów dotyczących potwierdzenia nadpisania istniejącego pliku. Gdybyśmy jednak chcieli zachować poprzedni plik z wyeksportowanymi danymi, należałoby sprawdzić, czy plik `dane.xlsx` istnieje w podanej lokalizacji i zapisać go pod zmienioną nazwą. Można tego dokonać w następujący sposób:

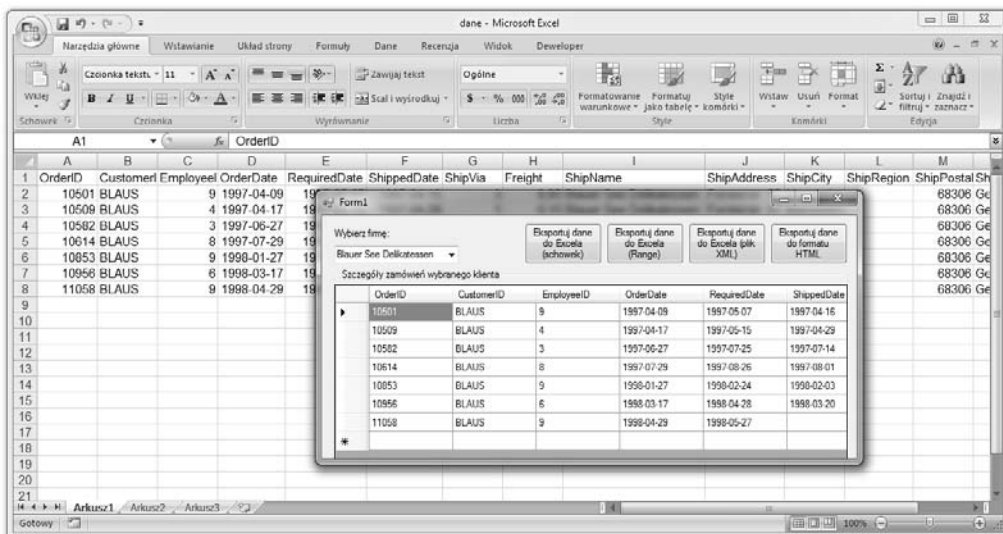
```
if (System.IO.File.Exists(sciezka))
    sciezka = sciezka.Replace(".xlsx", DateTime.Now.Second.ToString() + ".xlsx");
```

Eksport danych do formatu HTML

Ciekawym elementem metody `Workbook.SaveAs` jest możliwość zapisania pliku w różnych formatach. Pozwala to na stosunkowo prosty eksport danych do plików obsługiwanych przez zewnętrzne aplikacje. Omawiany przykład polega na zmodyfikowaniu poprzedniego projektu w taki sposób, aby eksportowane dane zostały zapisane w pliku `.html`. W tym celu wystarczy w metodzie `SaveAs` wskazać format pliku docelowego. Jeśli plikiem docelowym ma być jednoplikowa strona WWW, wywołanie metody `SaveAs` przyjmie następującą postać:

```
wb.SaveAs(sciezka, Excel.XlFileFormat.xlHtml);
```

Ostateczny wygląd aplikacji oraz przykładowy wynik jej działania przedstawiam na rysunku 12.2.



Rysunek 12.2. Aplikacja w trakcie działania

Skorowidz

- , 50
- !=, 50, 89, 90
- #define, 70
- #else, 70
- #endif, 70
- #endregion, 71
- #if, 70
- #region, 71
- .NET, 15, 45, 46
 - CAS, 47
 - CLR, 46, 616
 - CLS, 46
 - CTS, 46
 - IL, 46
 - JIT, 47
 - kod pośredni, 46
 - kod zarządzany, 46
 - kompilacja, 46
 - MSIL, 46
 - podwójna kompilacja, 46
 - system typów, 46
 - środowisko uruchomieniowe, 46
 - Windows Forms, 46
- .NET Compact, 45
- .NET Framework Common Language Runtime, 616
- .NET Remoting, 520
- ?., 50
- [], 52, 73
- [DataContract], 524
- [ExportMetadata], 601
- [OperationContract], 524
- [ServiceContract], 524
- [STAThread], 28
- [WebMethod], 524
- [WebService], 524
- ++, 50
- <, 89
- <<, 50
- <=, 89
- =, 47
- ==, 50, 89, 90
- =>, 60
- >, 89
- >=, 89
- >>, 50

A

- Abort, 452, 453
- AcceptChanges, 216
- Access, 311
- Access 2007, 312
- Action, 470
- ActiveCell, 652, 658
- ActiveSheet, 652
- ActiveWorkbook, 652
- ActiveX, 155, 650, 671
 - CLSID, 671
 - GUID, 671
 - odtworzenie plików multimedialnych, 671
 - Windows Media Player, 671
- ActiveX Data Objects, 206, 650
- Add, 27, 161
- Add New Data Source, 208
- AddPermission, 273
- AddRef, 649
- ADO, 206, 650
- ADO.NET, 46, 205, 206, 261, 311
 - Command, 206
 - Connection, 206, 208
 - DataAdapter, 206
 - DataReader, 206
 - DataRelation, 246
 - DataRow, 214
 - DataSet, 206, 209
 - dostęp do danych, 206
 - edycja rekordów, 215
 - integralność danych, 224
 - kontrola wprowadzanych danych, 216
 - modyfikacja danych w obiekcie DataSet, 211
 - parametryzacja kwerend, 221
 - połączenie ze źródłem danych, 206, 207
 - przesył danych, 206
 - reguły integralności danych, 227
 - szyfrowanie połączenia, 265
 - TableAdapter, 206, 221
 - TransactionScope, 261
 - tworzenie rekordów, 211
 - usuwanie danych z zachowaniem reguł integralności, 227
 - usuwanie rekordów, 220

- ADO.NET
 - wersjonowanie obiektu DataRow, 216
 - wstawianie danych, 228
 - XML, 343
- ADO.NET 2.0, 401
- ADO.NET Entity Data Model, 402
- adres URL, 786
- AdventureWorks, 775, 779
- AfterLabelEdit, 224
- Aggregate, 506, 507
- AggregateException, 477, 517
- agregacje, 506
 - kolekcje współbieżne, 507
- aktualizacja danych, 319
- aliasy, 96
- AllowDrop, 138
- ALTER, 275
- analiza drzewa wyrażenia, 361
- analiza plików XML, 339
- analiza zawartości biblioteki załadowanej
 - dynamicznie, 588
- Anchor, 23, 24
- animacja piłki, 721
- AnnouncementEndpoint, 562
- anulowanie breakpointu, 39
- aplikacje .NET, 46
- aplikacje bazodanowe, 275
- aplikacje konsolowe, 450, 767
 - czekanie na akceptację użytkownika, 769
 - catalogi specjalne zdefiniowane w bieżącym profilu użytkownika, 772
 - komunikat „okienkowy”, 770
 - lista dysków logicznych, 773
 - Main, 768
 - odczytywanie danych z klawiatury, 769
 - odczytywanie zmiennych środowiskowych, 773
 - okno dialogowe, 771
 - projekt, 767
 - punkt wejściowy, 768
 - strumienie, 769
 - wyświetlanie napisów, 768
 - zmiana strumienia, 770
- aplikacje okienkowe, 20, 101
- aplikacje WCF, 523
- aplikacje Web, 538
- aplikacje Win32, 609
- aplikacje XNA, 705
- app.config, 127
- AppDomain, 272
- Append, 54
- Application, 107
- application domain, 681
- Application Domain, 271
- application settings, 126
- Application.Exit, 107
- Application.Run, 37, 616
- application-level customizations, 681
- architektura klient-serwer, 649
- architektura WCF, 521
- argumenty, 57
 - argumenty nazwane, 58
 - argumenty opcjonalne, 57
 - przekazywanie przez referencje, 59
 - przekazywanie przez wartości, 58
- Array, 72
- Array.Sort, 74
- as, 50, 51
- AS, 411
- ascending, 417
- AsEnumerable, 316
- AsParallel, 470, 508, 510, 511
- AsParallel<T>, 418
- assembly, 681
- Assembly, 616
- Assembly.GetType, 591
- AssemblyInfo.cs, 28
- Assert, 792
- Assert.AreEqual, 793
- Association, 407, 408
- AssociationSet, 406, 407, 408
- AssociationSetMapping, 409
- ataki SQL Injection, 275
- atrybuty, 71, 336
 - Conditional, 72
 - DllImport, 72
 - MTA Thread, 72
 - Obsolete, 72
- AttachedToParent, 485
- attribute-based mapping, 318
- AudioEngine, 742
- AutoGenerateColumns, 239
- automatyczne śledzenie zmian w katalogu, 173
- automatyczne zarządzanie transakcjami, 261
- automatyzacja, 249, 649, 650
 - ActiveCell, 658
 - adres aktywnej komórki Excela, 657
 - CheckSpelling, 662
 - COM, 649
 - drukowanie, 662
 - edycja danych w komórkach Excela, 656
 - eksplorowanie danych w arkuszu kalkulacyjnym, 654
 - Excel, 650, 651
 - FontDialog, 665
 - formatowanie zaznaczonego fragmentu tekstu
 - w dokumencie Worda, 665
 - funkcje Excela, 659
 - Internet Explorer, 665
 - kopiowanie zawartości dokumentu Worda
 - do komponentu RichTextBox, 664
 - modyfikacja zawartości aktywnej komórki Excela, 657
 - Navigate, 665
 - obiekt GC, 654
 - obsługa zdarzeń serwera automatyzacji, 658
 - okna dialogowe serwera automatyzacji, 655
 - OpenFileDialog, 662
 - otwieranie dokumentu Worda, 661
 - pobieranie informacji o aktualnie uruchomionej aplikacji Excel, 651
 - PrintOut, 663
 - PrintPreview, 663
 - projektowanie serwera automatyzacji, 667
 - Range, 664
 - rejestracja zdarzeń, 658
 - Selection, 664

serwer, 650
 serwer automatyzacji Internet Explorer, 665
 serwer automatyzacji MS Excel, 650
 serwer automatyzacji MS Word, 661
 SheetChange, 658
 SheetSelectionChange, 658
 sprawdzanie pisowni, 662
 tworzenie dokumentu Worda, 661
 uruchamianie aplikacji MS Word, 661
 uruchamianie procedury serwera automatyzacji, 669
 uruchamianie serwera automatyzacji
 MS Excel, 652
 uzyskiwanie referencji do instancji aplikacji MS
 Excel, 654
 WebBrowser, 665
 Word, 661
 Word.Application, 661
 wstawianie tekstu do bieżącego dokumentu Worda, 663
 wywoływanie funkcji Worda, 662
 zamykanie serwera automatyzacji MS Excel, 652
 zapisywanie danych w pliku Excela, 655
 zapisywanie danych z wykorzystaniem okna
 dialogowego aplikacji klienckiej, 656
 zapisywanie dokumentu Worda, 663
 zaznaczanie i kopiowanie tekstu dokumentu
 do schowka, 664
 zmiana własności obiektu reprezentującego aktywną
 komórkę, 658
 AutoResetEvent, 466
 AutoSize, 121
 Average, 307

B

BackColor, 30, 101, 147
 BackgroundImage, 120
 BackgroundWorker, 119, 120, 449, 451, 672
 BackgroundWorker.DoWork, 119
 balloon, 104
 balloon tip, 124
 BalloonTipTitle, 124
 balon, 124
 banki dźwięków, 739
 banki wave'ów, 739
 Barrier, 466
 BasicHttpBinding, 524, 525, 526
 BasicHttpContextBinding, 524, 525
 baza danych, 46, 205
 Access 2007, 312
 ADO.NET, 206
 AdventureWorks, 775, 779
 bezpieczeństwo, 265
 edycja rekordów, 215
 integralność danych, 224
 klucze, 205
 klucze obce, 205
 klucze podstawowe, 205
 kontrola wprowadzanych danych, 216
 Northwind, 207
 pliki, 312
 poła, 205

połączenie ze źródłem danych, 207
 reguły integralności danych, 225
 rekordy, 205
 relacje, 205
 SQL Server, 257
 szyfrowanie połączenia, 265
 tabele, 205
 TableAdapter, 221
 transakcje, 257
 tworzenie rekordów, 211
 usuwanie rekordów, 220, 224, 320
 więzy integralności, 226
 wstawianie rekordów, 228, 320
 BeginEdit, 217
 bezpieczeństwo aplikacji bazodanowych, 265
 biblioteka POCO, 441
 biblioteki DLL, 65, 156, 175, 583, 616, 618
 dołączanie do projektu, 586
 ładowanie, 583
 ładowanie dynamiczne, 587
 ładowanie statyczne, 586
 MEF, 595
 typ dynamic, 595
 wtyczki, 595
 wywołanie funkcji, 586
 biblioteki kontrolek, 175
 biblioteki typu, 650
 BinaryExpression, 362
 binding, 524
 Binding, 236
 BindingContext, 234
 BindToMoniker, 654
 Bitmap, 148
 bitmapy, 131
 BlockingCollection, 501, 502
 BlockingCollection<>, 500
 bloki, 71
 błędy, 35
 błędy krytyczne, 620
 bool, 48, 756
 Boolean, 48
 BorderStyle, 121
 boxing, 63
 break, 67, 68, 79
 Breakpoint, 38
 Bring to Front, 105
 buforowanie grafiki, 131
 Business Intelligence, 277
 byte, 47, 48, 51, 54
 Byte, 48
 bytecode, 46

C

C#, 15, 45
 atrybuty, 71
 dyrektywy preprocesora, 70
 kolekcje, 72
 sterowanie przepływem, 66
 typy danych, 47
 wyjątki, 68

- C++, 45, 615
 - Call Stack, 41
 - CallingConvention, 634
 - CallingConvention.Cdecl, 634
 - Cancel, 475, 477
 - CancellationToken, 490
 - CancelEdit, 216
 - CancellationToken, 475, 476, 480, 482, 490, 515
 - CancellationToken.ThrowIfCancellationRequested, 475, 477
 - CancellationTokenSource, 475, 476, 515
 - CancellationTokenSource.Cancel, 475
 - Cancelled, 478
 - CAS, 47, 270, 681
 - case, 66
 - catch, 42, 68
 - CCW, 649
 - CDATA, 336
 - Centrum Instalacji SQL Server, 775
 - certyfikaty, 266
 - char, 48, 53
 - Char, 48
 - CharSet, 628
 - Chart, 237, 294
 - CheckBox, 217
 - checked, 50
 - Checked, 108
 - CheckSpelling, 662
 - CheckState, 108
 - chunk partitioning, 513
 - ciąg znaków połączenia, 402
 - ciągi definiujące łańcuchy, 53
 - class, 25, 61, 84, 190
 - Class Library, 156, 584, 667
 - CLI, 681
 - Click, 169
 - Clipboard.SetDataObject, 253
 - Close, 33, 107, 123
 - Close Solution, 24
 - Closed, 150
 - CloseMainWindow, 133
 - CLR, 46, 252, 619, 649, 681
 - CLS, 46
 - CLSID, 576, 671
 - Code Access Security, 47, 270, 681
 - Collect, 654
 - CollectionBag, 500
 - ColorDialog, 112
 - ColorProgressBar, 183, 187, 200
 - COM, 65, 519, 649
 - AddRef, 649
 - IUnknown, 649
 - klient, 649
 - pojemnik, 650
 - QueryInterface, 649
 - Release, 649
 - serwer, 649
 - COM Callable Wrapper, 649
 - COM+, 519, 649
 - ComboBox, 233, 234, 577
 - Command, 206
 - CommandLine, 772
 - Common Language Runtime, 46, 649
 - Common Language Specification, 46
 - Common Type System, 46
 - CompareTo, 74, 91, 94
 - Complex, 190, 790
 - Component Object Model, 649
 - Concat, 308
 - ConcurrentBag, 503
 - ConcurrentBag<>, 500
 - ConcurrentDictionary, 503
 - ConcurrentQueue, 482, 501, 503
 - ConcurrentQueue<>, 500
 - ConcurrentStack, 501, 503
 - ConcurrentStack<>, 500
 - Conditional, 72
 - ConfigurationManager, 270
 - ConfigurationManager.ConnectionStrings, 270
 - Connection, 206, 208
 - ConnectionString, 208, 240, 265, 266, 268, 274
 - Integrated Security, 274
 - ConnectionStringBuilder, 274
 - Console, 767
 - Console Application, 450, 767
 - Console.In, 769
 - Console.In.ReadLine, 769
 - Console.Out, 769
 - Console.Out.WriteLine, 769
 - Console.ReadLine, 769
 - Console.WriteLine, 769, 770
 - ConstantExpression, 362
 - Contains, 308
 - ContextMenuStrip, 122, 152, 225
 - continue, 67, 68
 - ContinueWhenAll, 480, 481
 - ContinueWhenAny, 480, 481
 - ContinueWith, 474, 475
 - Control, 173
 - CopyTo, 504
 - Count, 504
 - CountdownEvent, 466
 - CREATE, 275
 - Created, 478
 - CreateDatabase, 321
 - CreateProcess, 606
 - CTS, 46
 - CurrentDirectory, 772
 - cykl życia gry XNA, 709
 - czcionki, 112, 748
 - czekanie na ukończenie pracy wątku, 458
- ## D
- dane w programach równoległych, 499
 - Data Definition Language, 275
 - DataAdapter, 206
 - Database Management System, 205
 - DataBinding, 233, 234, 275
 - DataContext, 317, 319, 331
 - DataGrid, 307
 - DataGridView, 239, 243, 250, 301, 312, 325
 - Alignment, 243
 - brakujące dane, 243
 - czcionki, 243

- DefaultCellStyle, 243
- eksport danych, 254
- formatowanie danych, 240
- formularz z podformularzem, 246
- justowanie zawartości komórki, 243
- kolejność wyświetlanych kolumn, 242
- kolumny, 240
- prezentacja danych, 239
- tryby wyświetlania danych, 239
- wyróżnianie danych, 242
- DataGridViewCollection, 239
- DataGridViewContentAlignment, 243
- DataReader, 206
- DataRelation, 246
- DataRow, 214
 - BeginEdit, 218
 - CancelEdit, 216
 - Delete, 220
 - EndEdit, 216
 - HasVersion, 217
- DataSet, 206, 208, 209, 211, 239, 275, 311, 401
 - konfiguracja, 312
 - modyfikacja danych, 211
 - przenoszenie danych do pliku XML, 350
 - silna typizacja, 209, 210
 - słaba typizacja, 209
 - wczytywanie danych z pliku XML, 344
 - WriteXML, 255
 - zapisywanie zmian do pliku XML, 345
- DataSource, 239
- DataTable, 316
- DateTime.Now.Ticks, 126
- DBMS, 205
- DCOM, 519, 649
- DDL, 275
- Debug, 37
- DEBUG, 70
- debugger, 36
- debugowanie, 35, 36
 - Breakpoint, 38
 - Call Stack, 41
 - klawisze skrótów, 37
 - Locals, 39
 - obserwowanie wartości zmiennych, 39
 - punkt przerwania, 38
 - Run to Cursor, 38
 - śledzenie wykonywania pętli, 39
 - śledzenie wykonywania programu krok po kroku, 37
 - Watch, 39
- decimal, 48
- Decimal, 48
- default, 48
- DefaultCellStyle, 243
- DefaultCellStyle.NullValue, 243
- definicja delegacji, 59
- definicja uchwytu do metody, 60
- definicja wyrażenia lambda, 60
- definicje operatorów, 88
- definiowanie
 - aliasy, 96
 - metody, 55
 - metody anonimowe, 60
 - stałe preprocesora, 70
 - typy danych, 84
 - typy ogólne, 93
 - typy parametryczne, 92
 - typy wyliczeniowe, 54
 - zdarzenie, 187
 - zdarzenie wywołane po zmianie zaznaczonego elementu, 169
- deklaracja
 - tablice, 72
 - zmiennne, 47
- deklaracja XML, 335
- delegacje, 59
- delegate, 59, 60, 98, 169
- delete, 62
- DeleteDatabase, 321
- DeleteObject, 421
- DeleteSubKey, 575
- dequeue, 501
- descending, 417
- deserializacja obiektu z pliku XML, 343
- Deserialize, 343
- diagram związków encji, 434
- Dictionary, 77
- Directory, 159
- Directory.GetCurrentDirectory, 159
- Directory.GetDirectories, 162
- Directory.GetFiles, 162
- DirectoryEntry, 773
- DirectX, 755
- discovery, 558, 560
 - adres serwisu, 561
 - konfiguracja klienta, 561
 - programowe dodanie funkcji discovery do hosta, 560
 - punkt końcowy serwisu, 560
 - wywołanie serwisu, 561
- DiscoveryEndpoint, 562
- DisplayAlerts, 256, 656
- Dispose, 27, 616
- Distinct, 308
- Distributed COM, 649
- Distributed Component Object Model, 519
- DLL, 65, 156, 583
- DllImport, 72, 604, 618, 628, 634
- DllNotFoundException, 626
- DLR, 46
- do..while, 67
- Dock, 24, 101, 102, 105, 121, 132, 136, 157, 612
- document-level customizations, 681
- dodatki dla pakietu Office, 682
- dodatki na poziomie aplikacji, 681, 696
- dodawanie referencji do biblioteki systemowej platformy .NET, 584
- dodawanie referencji do Excela, 651
- DoDragDrop, 135
- dokowanie palety komponentów Toolbox, 22
- dokumenty XML, 335
 - atrybuty, 336
 - CDATA, 336
 - deklaracja, 335
 - elementy, 336
 - encje, 336

- dokumenty XML
 - komentarze, 336
 - XmlTextReader, 336
 - XmlTextWriter, 336
 - dołączanie bibliotek DLL użytkownika do projektu, 586
 - domeny aplikacji, 271, 681
 - domyślne wartości argumentów metod, 57
 - dostęp do danych, 206
 - dostęp do składowej, 50
 - double, 47, 48, 756
 - Double, 48
 - DPAPI, 268
 - DpapiProtectedConfigurationProvider, 268
 - drag & drop, 135
 - DragDrop, 135, 138
 - DragDropEffects.Copy, 138
 - DragDropEffects.Move, 138
 - DragOver, 135, 138, 139
 - Draw, 711
 - DrawableGameComponent, 732
 - DrawString, 113
 - DrawText, 113
 - Drillthrough, 297
 - drill-through report, 295
 - DriveInfo, 606
 - DROP, 275
 - drukowanie, 113
 - drukowanie w tle, 119
 - okno dialogowe ustawień wydruku, 116
 - PageSetupDialog, 116
 - podgląd wydruku, 116
 - PrintDocument, 114
 - PrintPreviewControl, 117
 - problem długich linii, 118
 - tekst, 114
 - Drukuj..., 114
 - drzewa wyrażenia, 303, 354, 361
 - drżenie ekranu, 764
 - duplex, 525
 - duszek, 708, 713
 - dwukrotne kliknięcie, 163, 171
 - dymki, 104, 124
 - dynamic, 49, 63, 65, 595, 682
 - Dynamic Language Runtime, 46
 - dynamic link library, 583
 - dynamiczna zmiana typu zmiennej, 63
 - dynamiczne ładowanie biblioteki DLL, 583, 587
 - dynamiczne rozpoznawanie typów, 587
 - dyrektywy preprocesora, 70
 - #define, 70
 - #else, 70
 - #endif, 70
 - #endregion, 71
 - #if, 70
 - #region, 71
 - bloki, 71
 - kompilacja warunkowa, 70
 - stałe preprocesora, 70
 - dyski, 159, 161
 - dyski logiczne, 773
 - dziedziczenie typu tabela na hierarchię, 434
 - dziedziczenie typu tabela na typ, 438
 - dźwięki, 120, 125, 739
 - dźwięki systemowe, 604
- ## E
- Eager loading, 421, 423
 - Edit Relation, 226
 - EDM, 401
 - edmx, 405
 - edycja, 113
 - dane w komórkach Excela, 656
 - dane w źródle LINQ, 393
 - dźwięk, 739
 - menu kontekstowe, 153
 - metoda zdarzeniowa, 32
 - rekord, 215
 - edytor menu, 105
 - Edytor rejestru, 577
 - efekty HLSL, 755, 758, 759, 762
 - drżenie ekranu, 764
 - falowanie obrazu, 764
 - inwersja kolorów obrazu, 762
 - negatyw obrazu, 763
 - zanikanie obrazu, 762
 - InvalidOperationException, 264
 - ekran powitalny, 120
 - eksplorowanie danych w arkuszu kalkulacyjnym, 654
 - eksport danych, 249
 - format HTML, 256
 - macierz danych, 249
 - pliki tekstowe, 249
 - pliki XML, 250, 254
 - raporty, 297
 - schowek systemowy, 250
 - wstawianie danych do arkusza kalkulacyjnego, 253
 - ElapsedTime, 710
 - elementy XML, 336
 - else, 66
 - Enabled, 122
 - EnableEventRising, 173
 - encje, 401
 - EndEdit, 216, 219
 - EndsWith, 52
 - EnforceConstraints, 227
 - enqueue, 501
 - entity class, 317
 - Entity Framework, 401
 - AddTo, 419
 - aktualizacja modelu z bazy danych, 429
 - Association, 404
 - atrybuty, 401
 - ciąg znaków połączenia, 402
 - Create, 419
 - DeleteObject, 421
 - dodawanie obiektów encji, 427
 - dziedziczenie, 434
 - dziedziczenie typu tabela na hierarchię, 434
 - dziedziczenie typu tabela na typ, 438
 - EDM, 401, 404
 - encje, 401

- Entity SQL, 409
 - filtrowanie danych, 411
 - grupowanie danych, 414
 - klasa kontekstu, 443
 - kontekst, 409
 - leniwe ładowanie, 421, 424
 - LINQ to Entities, 409
 - mapowania procedur składowanych, 426
 - mapowanie procedur składowanych pobierających dane, 428
 - metadane, 403, 405
 - model danych encji, 401
 - model konceptualny, 406
 - model związków encji, 401
 - model źródła danych, 407
 - modyfikacja encji, 424
 - modyfikacja modelu danych encji, 432
 - modyfikacja obiektów encji, 419, 420, 445
 - ObjectContext, 409
 - ObjectServices, 409
 - plik .edmx, 405
 - pobieranie danych, 428
 - pobieranie obiektów encji, 444
 - POCO, 441
 - połączenie ze źródłem danych, 403
 - procedury składowane, 424
 - procedury składowane modyfikujące encje, 424
 - procedury składowane pobierające dane, 428
 - programowanie równoległe, 418
 - projekcja danych, 412
 - projekt, 402
 - relacje, 404
 - SaveChanges, 419
 - sortowanie danych, 417
 - tworzenie obiektów encji, 419
 - tworzenie tabel w bazie danych na podstawie modelu encji, 431
 - usuwanie obiektów encji, 419, 421, 428
 - zachłanne ładowanie, 421, 423
 - zapytania, 409
 - zapytania Entity SQL, 411
 - zapytania LINQ to Entities, 410
 - związki, 401
 - związki między encjami w zapytaniach, 415
 - źródło danych, 407
 - Entity Relationship Model, 401
 - Entity SQL, 409, 410, 411, 413
 - GROUP BY, 415
 - grupowanie danych, 415
 - HAVING, 415
 - ORDER BY, 417
 - relacje pomiędzy encjami, 416
 - SELECT, 413
 - sortowanie danych, 417
 - EntityContainer, 406
 - EntityContainerMapping, 409
 - EntityObject, 441
 - EntitySet, 406, 408
 - EntitySetMapping, 409
 - EntityType, 406
 - entry point, 21, 28
 - EntryPoint, 628
 - EntryPointNotFoundException, 620
 - Enum, 54
 - Environment, 617, 772, 773
 - Environment.GetFolderPath, 772
 - Environment.GetLogicalDrives, 774
 - Environment.SpecialFolder, 772
 - Equals, 52, 89, 90
 - ERD, 434
 - ERM, 401
 - Esc, 151
 - etykiety, 284
 - event, 59, 169
 - EventWaitHandle, 465, 466
 - ExactSpelling, 628
 - Excel, 249, 652
 - automatyzacja, 650
 - Excel.Application, 252, 652
 - Excel.GetSaveAsFilename, 655
 - Excel.Range, 658
 - Exception, 41, 68
 - exe, 46
 - Execute, 356
 - ExecuteCommand, 321
 - ExecuteQuery, 321
 - Exit, 107
 - ExpectedExceptionAttribute, 794
 - explicit, 91, 195
 - ExportMetadata, 601
 - Expression, 358
 - expression tree, 361
 - ExpressionVisitor, 362
 - extension methods, 98
 - extern, 604
- F**
- fabryka zadań, 480
 - tworzenie, 482
 - falowanie obrazu, 764
 - Faulted, 478
 - FIFO, 77, 502
 - FileListBox, 156, 157, 160, 179
 - FileName, 112
 - FileNotFoundException, 109
 - FileSystemWatcher, 173
 - FillRectangle, 129
 - FILO, 77
 - Filter, 111
 - filtrowanie danych, 162, 289
 - filtrowanie raportu na etapie pobierania danych źródłowych, 290
 - finally, 69
 - FindWindow, 610
 - float, 47, 48, 756
 - FontDialog, 112, 665
 - for, 67, 150
 - pętla równoległa, 487
 - For, 487, 493
 - ForAll, 517
 - foreach, 68, 74
 - pętla równoległa, 489

ForEach, 470, 489
 ForeColor, 101
 Form1.cs, 21
 Form1.Designer.cs, 21
 formatowanie komórek, 693
 formatowanie zaznaczonego fragmentu tekstu
 w dokumencie Worda, 665
 FormBorderStyle, 121, 147, 577
 FormClose, 107, 684
 FormClosed, 128
 FormClosing, 107
 FormLoad, 684
 formy, 21, 102
 fraktale, 198, 201
 from, 309
 FromArgb, 32
 FullName, 49, 99
 funkcje haszujące, 513
 funkcje HLSL, 758
 funkcje WinAPI, 603
 CreateProcess, 606
 FindWindow, 610
 GetDiskFreeSpaceEx, 606
 GetVersionEx, 629, 630, 631
 GetWindowsDirectory, 608
 LoadLibrary, 625
 MessageBeep, 604
 MessageBox, 627
 MessageBoxA, 627
 MessageBoxW, 627, 628
 SendMessage, 609, 611
 ShellExecute, 606
 wersje funkcji, 627
 WinExec, 605
 zwracanie tablicy znaków, 608

G

GAC, 529
 GameComponent, 732
 gamepad, 727, 729
 GamePadState, 729
 GameServices, 733
 GameTime, 710
 garbage collector, 27, 54, 62, 65, 569, 616, 629
 GarbageCollector, 654
 Gauge, 292
 GC, 654
 GDI, 277
 generator liczb pseudolosowych, 66, 456, 496
 generic types, 75, 92
 generowanie grafiki 2D, 713
 GeometryServiceLibrary, 536
 GetActiveObject, 252, 651, 652
 GetConsumingEnumerable, 502, 506
 GetCurrentDirectory, 159
 GetCurrentThreadId, 621
 GetDirectories, 159, 162
 GetDiskFreeSpaceEx, 606
 GetEnumerator, 354, 504
 GetEnvironmentVariable, 773

GetEnvironmentVariables, 773
 GetFileName, 160
 GetFiles, 159, 162
 GetHashCode, 89, 90
 GetLogicalDrives, 159, 774
 GetParameters, 590
 GetProcesses, 133, 610
 GetSaveAsFilename, 655
 GetScheduledTasks, 483
 GetSystemDirectory, 608
 GetSystemInfo, 591
 GetSystemInfoBuilder, 591
 GetSystemMetrics, 141
 GetType, 49, 51, 63, 99, 587, 590, 591
 GetValueOrDefault, 62
 GetVersion, 617, 619, 621
 GetVersionEx, 617, 629, 630, 631
 GetWindowsDirectory, 608
 Global Assembly Cache, 529
 Globally Unique Identifier, 671
 główne wejście programu, 28
 gniazda, 520
 graficzne wskaźniki, 292
 graficzny interfejs użytkownika, 19, 101
 grafika, 129
 grafika 2D, 705
 grafika 3D, 755
 graphical user interface, 101
 Graphics, 113, 129, 130, 182
 Graphics Device Interface, 277
 Graphics.DrawImage, 131
 GraphicsDeviceManager, 708
 GreetingServiceHosts.Tcp, 563
 GreetingServiceLibrary, 563
 GROUP BY, 415
 grupa przycisków, 73
 grupowanie danych, 414
 gry XNA, 707
 GUI, 20, 101
 GUID, 671

H

half, 756
 hash partitioning, 513
 HasMorePages, 115
 HasShutdownStarted, 772
 HasVersion, 217
 HAVING, 415
 Height, 129
 Helion.SystemInfo.GetInfo, 591
 HLSL, 756
 definiowanie funkcji, 758
 efekty, 759
 funkcje, 758
 macierze, 756
 próbkiwanie tekstur, 756
 sampler, 756
 semantyka, 757
 semantyki wejściowe shaderów pikseli, 757
 semantyki wejściowe shaderów wierzchołków, 757

- semantyki wyjściowe shaderów pikseli, 758
 - semantyki wyjściowe shaderów wierzchołków, 757
 - sterowanie przepływem, 758
 - tex2D, 761
 - typy danych, 756
 - wektory, 756
 - HorizontalScrollbar, 132
 - hosting, 534
 - HTML, 256
 - HTTP, 266, 520, 525
 - Hypertext Transfer Protocol, 520
- I**
- IComparable, 74, 75, 91, 94
 - CompareTo, 91
 - Icon, 103
 - IcontractBehavior, 551
 - IConvertible, 86
 - IDENTITY, 420
 - identyfikacja aplikacji, 609
 - IDictionary, 77
 - IDirectory, 773
 - IDispatch, 667
 - IDisposable, 410
 - IEMonitor, 640
 - IEMonitorSetup, 644
 - IendpointBehavior, 551
 - IEnumerable, 79, 303, 305, 353, 483
 - IEnumerable<>, 355, 500, 507
 - IEnumerator, 354
 - if, 66
 - if..else, 66
 - IFormattable, 191
 - IGeometryService, 534
 - IIS, 538
 - ikona aplikacji, 103
 - ikona okna, 103
 - ikony biblioteki, 177
 - ikony w obszarze powiadamiania, 122
 - IL, 46, 616
 - ILinqToTxt, 377, 384
 - ilość wolnego miejsca na dysku, 606
 - imię żeńskie, 785
 - implementacja interfejsu, 91
 - IComparable, 91
 - implementacja przez typ ogólny, 95
 - implicit, 91
 - implicitly-typed variables, 49
 - import funkcji z biblioteki systemowej, 617
 - import wielu funkcji, 621
 - klasa opakowująca, 621
 - ładowanie biblioteki DLL, 625
 - wersja z zabezpieczeniem, 619
 - IncludeSubdirectories, 173
 - indeksator, 52
 - IndexOf, 52
 - informacje o platformie .NET, 585
 - informacje o profilu użytkownika, 771
 - informacje o systemie, 585, 767, 771
 - informacje o środowisku aplikacji, 771
 - inicjacja
 - liczby, 49
 - obiekty, 80
 - tablice, 80
 - zmienne, 47
 - InitialDirectory, 111, 112
 - InitializeComponent, 37, 160
 - InitializeComponents, 25, 27
 - inline, 29
 - InnerExceptions, 477
 - Insert, 52, 53, 54
 - inspekcja metody, 592
 - instalacja
 - baza danych AdventureWorks, 775, 779
 - baza danych Northwind, 207
 - SQL Server 2008 Express Edition, 775
 - usługi Windows, 641
 - Installed Templates, 20
 - installutil.exe, 637
 - instancje klasy, 62, 83
 - int, 47, 48, 756
 - Int16, 48
 - Int32, 48
 - Int64, 48
 - integracja kodu zarządzanego i niezarządzanego, 615
 - integracja projektu zarządzanego i niezarządzanego, 632
 - integralność danych, 224
 - Integrated Performance Primitives, 616
 - Integrated Security, 274
 - Intel IPP, 616
 - IntelliSense, 30
 - interakcja z komponentem, 168
 - interfejs aplikacji, 20
 - interfejsy, 91
 - IComparable, 74, 91
 - IConvertible, 86
 - IDictionary, 77
 - IDispatch, 667
 - IEnumerable, 305, 353
 - IEnumerable<>, 355
 - IFormattable, 191
 - ILinqToTxt, 377
 - implementacja, 91
 - IOrderedQueryable<>, 360
 - IProducerConsumerCollection, 499
 - IProducerConsumerCollection<>, 503
 - IQueryable, 354, 359
 - IQueryable<>, 360
 - IUnknown, 649
 - Interlocked, 494, 499
 - Intermediate Language, 46, 616
 - Internet Explorer, 665
 - Interoperability, 617
 - Intersect, 309
 - IntPtr, 610, 634
 - InvalidOperationException, 91
 - Invoke, 489, 593
 - inwersja kolorów obrazu, 762
 - IoperationBehavior, 551
 - IOrderedQueryable, 359
 - IOrderedQueryable<>, 360, 367
 - IProducerConsumerCollection, 482, 499, 503

IProducerConsumerCollection<>, 499, 500, 503, 504
 IQueryable, 303, 354, 359
 IQueryable<>, 360, 367
 IQueryProvider, 303, 358, 384
 is, 50, 51
 IS NOT NULL, 415
 IS NULL, 415
 IsBackground, 452
 IsCancellationRequested, 475
 IServiceBehavior, 551
 IsExceptional, 491
 IsFixedTimeStep, 710
 IsKeyDown, 727
 IsMouseVisible, 728
 IsPublic, 590
 IsStatic, 590
 IsStopped, 491
 IsSynchronized, 504
 IUnknown, 649

J

J++, 45
 jeden do jednego, 404
 jeden do wielu, 404
 język C#, 15, 45
 język HLSL, 756
 język IL, 616
 język pośredni, 616
 język RDL, 301
 język SQL, 317
 język T-SQL, 257, 275, 317
 język XML, 335
 JIT, 46, 47
 join, 309, 328
 Join, 458
 Just-In-Time, 46, 47

K

katalogi, 158, 159, 161
 katalogi specjalne zdefiniowane
 w bieżącym profilu użytkownika, 772
 prezentacja zawartości w komponencie, 160
 kernel32.dll, 617
 key, 77
 Key, 77
 Key Performance Indicators, 292
 Keyboard.GetState, 727
 KeyChar, 33
 KeyPress, 33, 151
 KeyPreview, 33, 131, 151
 Keys, 727
 Kill, 133
 klasa encji, 317
 klasa opakowująca, 621
 klasy, 24, 61, 65, 83, 84
 Action, 470
 Application, 107
 Array, 72
 Assert, 792

Barrier, 466
 BlockingCollection, 501
 CancellationToken, 475, 490, 515
 CancellationTokenSource, 475, 515
 ConcurrentQueue, 501
 ConcurrentStack, 501
 ConfigurationManager, 270
 Console, 767
 Control, 173
 DataContext, 317, 319
 DataTable, 316
 deklaracja, 25
 Directory, 159
 DpapiProtectedConfigurationProvider, 268
 DriveInfo, 606
 Enum, 54
 Environment, 617, 772
 EventWaitHandle, 465
 Exception, 41, 68
 ExpressionVisitor, 362
 GamePadState, 729
 Graphics, 113, 129, 182
 GraphicsDeviceManager, 708
 instancje, 62, 83
 Interlocked, 494, 499
 konstruktor, 84
 Lazy, 55, 602
 MarshallByRefObject, 272
 MediaPlayer, 745
 MethodInfo, 590
 metody, 55, 83
 metody statyczne, 85, 195
 Monitor, 465, 467
 Mutex, 467
 Object, 51
 Panel, 22
 Parallel, 447, 469, 487
 ParallelEnumerable, 507
 ParallelLoopResult, 491
 ParallelLoopState, 491
 ParallelOptions, 490
 ParameterizedThreadStart, 461
 Partitioner, 499
 pola, 83
 pola statyczne, 85
 Process, 133
 Queue, 77
 Random, 495
 RandomThreadSafe, 496
 Registry, 571, 578
 RegistryKey, 573
 RegularExpressions, 781
 RSACryptoProvider, 268
 ServiceController, 646, 648
 ServiceInstaller, 641
 ServiceProcessInstaller, 641
 SortedList, 77
 SpecialFolders, 571
 SpinLock, 466
 Sprite, 714
 SpriteBatch, 708, 748
 SqlConnection, 272

- Stack, 77
- StreamReader, 109, 179
- String, 52, 53
- StringBuilder, 53, 608
- SystemColor, 101
- SystemInfo, 585, 586, 595
- Task, 447, 469
- TaskFactory, 469, 480
- TaskScheduler, 469, 482
- Thread, 120, 451
- ThreadPool, 464
- ThreadStart, 451
- TimeSpan, 710
- TrackBar, 22
- TransactionScope, 261
- UserControl, 157, 168
- WebConfigurationManager, 270
- własności, 83
- XDocument, 346, 348
- XElement, 346
- XmlSerializer, 342
- XmlTextReader, 336, 339
- XmlTextWriter, 336, 338
- klasy specjalne, 54
- klasy statyczne, 28
- klawiatura, 33, 727, 769
- klawisze skrótów, 106
 - edytor Visual C# 2010, 31
- klient WCF, 542
- klucze, 77, 205
- klucze obce, 205
- klucze podstawowe, 205
- kluczowe wskaźniki efektywności, 292
- kod aplikacji, 24
- kod HLSL, 760
- kod klawisza, 33
- kod MSIL, 46
- kod nadzorowany, 616
- kod niezarządzany, 617
- kod pocztowy, 784
- kod pośredni, 46, 252
- kod zarządzany, 46, 252, 616, 617
- kodowanie znaków, 628
- kolejki, 77
- kolejki MSMQ, 527
- kolejkowanie zadań, 482
- kolejność przejmowania focusu
 - przez komponenty, 27
- kolekcje, 72
 - Dictionary, 77
 - kolejki, 77
 - List, 75
 - listy, 75
 - słowniki, 77
 - SortedDictionary, 77
 - SortedList, 77
 - stos, 77
 - System.Collections, 72
 - tablice, 72
- kolekcje współbieżne, 500
 - Aggregate, 507
 - agregacje, 506, 507
 - BlockingCollection, 501
 - CollectionBag, 500
 - ConcurrentQueue, 501
 - ConcurrentStack, 501
 - IEnumerable<>, 507
 - IProducerConsumerCollection<>, 503, 504
 - kolejka, 501
 - stos, 501
 - tworzenie kolekcji współbieżnej, 503
- kolizje, 721, 723
- kolorowy pasek postępu, 181
 - ColorProgressBar, 183
 - metody, 186
 - PerformStep, 186
 - pola, 182
 - rysowanie paska postępu, 185
 - własności, 182
 - zdarzenia, 186
- kolory, 29, 30, 32, 112
 - kolor tła, 30
- kolumny obliczeniowe raportu, 280
- komendy DDL, 275
- komentarze XML, 336
- kompilacja, 46
 - kompilacja komponentu
 - do postaci biblioteki DLL, 175
 - kompilacja warunkowa, 70
- kompilator JIT, 46, 47
- komponenty, 27, 155
 - BackgroundWorker, 451, 672
 - Chart, 237
 - CheckBox, 217
 - ComboBox, 577
 - ContextMenuStrip, 152
 - DataGridView, 239, 250
 - DataSet, 311
 - FileListBox, 156
 - FileSystemWatcher, 173
 - FontDialog, 665
 - kompilacja do postaci biblioteki DLL, 175
 - komponenty .NET, 155
 - komponenty gry, 731
 - komponenty niewidoczne, 155
 - ListBox, 132, 142
 - ListView, 223
 - modyfikacja własności, 32
 - odświeżanie zawartości, 173
 - OpenFileDialog, 677
 - pasek postępu, 181
 - ProgressBar, 181, 607
 - przypisywanie metody zdarzeniowej, 31
 - ReportViewer, 277, 278
 - RichTextBox, 664
 - SerialPort, 155
 - SoundPlayer, 120, 125
 - SplitContainer, 135
 - stan, 157
 - StatusStrip, 647
 - TextBox, 577
 - Timer, 122, 125, 155
 - tworzenie, 156

- komponenty Windows Forms, 22, 101, 684
 - MenuStrip, 105
 - StatusStrip, 104
- komunikacja między wątkami, 465
- komunikacja typu duplex, 525
- komunikacja typu „pytanie – odpowiedź”, 526
- komunikacja typu simplex, 525
- komunikaty o wyjątkach, 42
- komunikaty Windows, 609
 - identyfikacja aplikacji, 609
 - monitor komunikatów, 612
 - obsługa, 609
 - odbieranie, 612
 - porozumiewanie się z systemem, 611
 - PostMessage, 609
 - reakcja na wybrany komunikat, 613
 - SendMessage, 609, 611
 - WM_CLOSE, 611
 - WM_CTLCOLORLISTBOX, 612
 - WM_NCMOUSEMOVE, 613, 614
 - WM_SYSCOMMAND, 611
 - WndProc, 609, 612
 - wysyłanie, 609
 - wysyłanie komunikatu do okna o znanym uchwycie, 611
- konfiguracja
 - DataSet, 312
 - EDM, 402
 - formy, 147
 - pasek stanu, 104
 - reguły integralności danych, 226
 - szyfrowane połączenie ze źródłem danych, 265
 - WCF, 550
- konkurencyjność transakcji, 264
- konstruktory, 84
 - konstruktor domyślny, 85
- kontrakt, 524, 530
- kontrola głośności odtwarzania, 675
- kontrola transakcji, 257
- kontrola typów, 65
- kontrola wprowadzanych danych, 216
- kontrola wykonywania pętli, 491
- kontrolery gry, 727
- kontrolki, 22, 155
 - ActiveX, 155, 650, 671
 - ColorProgressBar, 183
 - kontrolki użytkownika, 156
 - obsługa, 29
 - Panel, 22
 - TrackBar, 22
 - własności, 164
- kontrolowane uruchamianie aplikacji, 36
- kontynuowanie zadań, 474
- konwersja kolekcji do pliku XML, 349
- konwersja na łańcuch, 86
- konwersja na typ double, 86
- konwersja pliku XML do kolekcji, 351
- konwersja tabeli bazy danych do pliku XML, 350
- konwersja typów podstawowych, 51
- konwersja typów zarządzanych i niezarządzanych, 628, 629

- kopiowanie
 - obszar pamięci kodu niezarządzanego, 635
 - zawartość dokumentu Worda do komponentu RichTextBox, 664
- Kreator formuł, 283
- kreator źródła danych, 325
- kwerendy SQL, 206

L

- Label, 120
- lambda, 60
- LambdaExpression, 362
- Language Integrated Query, 303
- LastIndexOf, 52
- Lazy, 55, 601, 602
- Lazy Loading, 421, 423
- Length, 52, 73
- leniwe inicjowanie zmiennych, 55
- leniwe ładowanie, 421, 423, 424
- liczby, 48
 - liczby losowe, 353
 - liczby pierwsze, 472
 - liczby pseudolosowe, 451
- liczby zespolone, 189
 - Complex, 190, 193
 - Imag, 191
 - metody, 191
 - operatory, 193
 - postać trygonometryczna, 192
 - projekt, 190
 - Real, 191
 - własności, 192
- LIFO, 77, 502
- LINQ, 45, 49, 65, 98, 303, 305
 - Average, 307
 - Concat, 308
 - Contains, 308
 - Distinct, 308
 - drzewa wyrażenia, 303
 - filtrowanie, 306
 - from, 309
 - grupowanie danych w zapytaniu, 308
 - IEnumerable, 305
 - Intersect, 309
 - join, 309
 - łączenie danych z różnych źródeł w zapytaniu, 309
 - łączenie zbiorów danych, 308
 - Max, 307
 - metody rozszerzające, 306
 - Min, 307
 - modyfikacja danych źródła, 309
 - operatory, 305
 - pobieranie danych, 306
 - prezentacja pobranych danych, 307
 - prezentacja w grupach, 308
 - select, 310
 - Single, 307
 - sortowanie, 306
 - Sum, 307

- Union, 308
 - weryfikacja danych, 308
 - wybór elementu, 307
 - wynik zapytania, 308
 - wyrazenia lambda, 307
 - zapytania, 303, 305, 307
 - źródła danych, 353
 - LINQ to DataSet, 303, 311, 313
 - filtrowanie, 316
 - obliczenia wykonywane na danych z tabeli, 316
 - orderby, 316
 - pobieranie danych, 314
 - sortowanie, 316
 - tworzenie tabeli, 314
 - where, 316
 - LINQ to Entities, 409, 410
 - AsParallel<T>, 418
 - filtrowanie danych, 411
 - grupowanie danych, 414
 - grupowanie danych z wykluczeniem wartości NULL, 414
 - leniwe ładowanie, 424
 - orderby, 417
 - programowanie równoległe, 418
 - projekcja danych, 412
 - select, 412
 - sortowanie danych, 417
 - where, 411
 - właściwości nawigacyjne, 415
 - zachłanne ładowanie, 423
 - zapytania równoległe, 418
 - związki między encjami w zapytaniach, 415
 - LINQ to Objects, 303, 305, 310
 - LINQ to SQL, 303, 317
 - aktualizacja danych, 319
 - Associations, 328
 - automatyczne tworzenie interfejsu użytkownika, 325
 - DataContext, 317, 319
 - dodawanie rekordów, 320
 - join, 328
 - klasa encji, 317
 - kreator źródła danych, 325
 - łączenie danych z tabel, 328
 - modyfikacja rekordów, 320
 - operacje, 321
 - pobieranie danych, 319
 - pobieranie danych za pomocą procedur składowanych, 331
 - procedury składowane, 331
 - projektowanie klasy encji, 322
 - relacje, 328
 - usuwanie rekordów, 320
 - współpraca z kontrolkami tworzącymi interfejs aplikacji, 325
 - LINQ to SQL Classes, 322
 - LINQ to TXT, 367
 - analizator zapytania, 377
 - dodawanie rekordów, 393
 - edycja danych w źródle LINQ, 393
 - ILinqToTxt, 384
 - List<string>, 384
 - mapowanie typów, 374
 - mapowanie własności, 385
 - metoda przetwarzająca dane, 385
 - modelowanie danych, 374, 386
 - odczytywanie pliku tekstowego, 370
 - pobieranie danych, 386
 - projekt źródła danych, 377
 - przetwarzanie danych z analizatora, 384
 - sortowanie, 386
 - stosowanie, 397
 - zapisywanie danych w instancjach klas encji, 392
 - LINQ to XML, 78, 303, 310, 336, 346
 - modyfikacja pliku XML, 351
 - pobieranie wartości z elementów o znanej pozycji w drzewie, 347
 - przenoszenie danych z bazy danych (komponentu DataSet) do pliku XML, 350
 - przenoszenie danych z kolekcji do pliku XML, 349
 - tworzenie pliku XML, 346
 - XDocument, 346, 348
 - XElement, 346
 - zapytania LINQ, 350
 - LINQ to XSD, 352
 - List, 75, 109, 286, 303
 - List<>, 72
 - lista argumentów wybranej metody, 591
 - lista dysków logicznych, 773
 - lista metod w klasie z biblioteki DLL, 589
 - lista odtwarzanych plików, 676
 - lista plików w katalogu, 158
 - lista rozszerzeń plików, 578
 - lista uruchomionych procesów, 132
 - ListBox, 132, 136, 142
 - SelectionMode, 142
 - ListView, 223
 - listy, 75, 284
 - Load, 338
 - LoadContent, 709
 - LoadLibrary, 625, 626
 - LocalAccount, 642
 - Locals, 39
 - Location, 27
 - lock, 461, 493
 - long, 48
 - LongRunning, 485
 - LowestBreakIteration, 491
 - LParam, 614
- L**
- ładowanie biblioteki DLL, 583, 625
 - ładodne znikanie okna, 150
 - łańcuchy, 52
 - ciągi definiujące łańcuchy, 53
 - String, 53
 - StringBuilder, 53
 - Unicode, 52
 - łączenie danych z różnych źródeł w zapytaniu LINQ, 309

M

- macierze, 287, 756
- Main, 21, 28, 768
- makra VBA, 681
- managed code, 46
- Managed Extensibility Framework, 595
- ManualResetEvent, 466
- mapowanie LINQ, 376
- mapowanie oparte na atrybutach, 318
- Mapping, 318
- Marshal, 651
- Marshal.BindToMoniker, 654
- Marshal.Copy, 635
- Marshal.GetLastWin32Error, 626
- MarshalByRefObject, 272
- Matrix, 287
- Max, 307
- MaxDegreeOfParallelism, 490, 513
- MaxValue, 48
- MB_ICONEXCLAMATION, 604
- MbUnit, 789
- mechanizm Garbage Collector, 629
- mechanizm PInvoke, 603
- mechanizm Refactor, 628
- mechanizm reflection, 65
- mechanizm transakcji, 257
- MediaPlayer, 739, 745
 - regulacja głośności, 745
- MEF, 595
 - definiowanie klasy gospodarza, 596
 - ExportMetadata, 601
 - gospodarz, 595
 - katalog, 599
 - metadane, 601
 - rozdzielenie gospodarza i wtyczki, 598
 - selekcja wtyczek, 602
 - tworzenie obiektu gospodarza, 597
 - tworzenie pojemnika wtyczek, 597
 - wiele wtyczek implementujących ten sam interfejs, 600
 - wtyczka, 595
- MemberExpression, 362
- Menedżer zadań, 457
- menu, 108
 - Edycja, 113
 - klawisze skrótów, 106
 - Plik, 114
 - polecenia, 108
 - zmiana stanu pozycji, 108
- menu główne, 101, 105
- Menu Item, 114
- menu kontekstowe, 30, 152
- menu kontekstowe plików *.config, 547
- MenuStrip, 105
- Message Queuing, 609
- MessageBeep, 603, 604
- MessageBox, 585, 621, 622, 627
- MessageBox.Show, 110, 325, 450, 770
- MessageBoxA, 627
- MessageBoxButtons, 622
- MessageBoxIcon, 622
- MessageBoxResult, 622
- MessageBoxW, 627, 628
- metadane, 616
- Metadata Exchange, 532
- metatypy, 83
- MethodCallExpression, 362
- MethodInfo, 590
- metoda Monte Carlo, 449, 492
- metody, 55, 83
 - argumenty, 57
 - argumenty nazwane, 58
 - argumenty opcjonalne, 57
 - Close, 107
 - CompareTo, 91, 94
 - definiowanie, 55
 - Dispose, 27
 - domyślne wartości argumentów, 57
 - Equals, 89, 90
 - GetHashCode, 89, 90
 - InitializeComponent, 37, 160
 - InitializeComponents, 27
 - Main, 21, 28, 768
 - metody anonimowe, 60
 - metody klasy, 29
 - metody rozszerzające, 98
 - metody statyczne, 85, 98, 195
 - metody z nieokreśloną liczbą argumentów, 78
 - parametry, 57
 - przeciążanie, 56
 - Refresh, 173, 185
 - tablice jako argumenty, 78
 - ToDouble, 86
 - ToString, 48, 86, 192
 - uchwyt, 60
 - wartości zwracane, 58
 - wywołanie, 55
 - zwracanie wartości, 58
 - zwracanie wartości przez argument, 58
- metody zdarzeniowe, 24, 29, 169
 - edycja, 32
 - przypisywanie do komponentów, 31
 - tworzenie, 29
 - wywoływanie z poziomu kodu, 32
- MEX, 532, 536
- Microsoft Cross-Platform Audio Creation Tool, 740
- Microsoft Intermediate Language, 252
- Microsoft Message Queuing, 519
- międzynarodowe wersje interfejsu, 28
- MIME type, 286
- Min, 307
- MinValue, 48
- mock objects, 794
- mock objects frameworks, 794
- Model Browser, 429, 430
- model danych encji, 401
- model związków encji, 401
- modelowanie danych, 318
- modelowanie danych ORM, 401
- modyfikacja
 - dane w obiekcie DataSet, 211
 - łańcuchy znakowe, 787
 - pliki XML, 351
 - własności komponentów, 32
 - wstążka, 687

modyfikatory
 delegate, 169
 event, 169
 explicit, 91, 195
 extern, 604
 implicit, 91
 override, 173
 private, 84, 151
 public, 84
 static, 85, 604
 Monitor, 465, 467
 monitor komunikatów, 612
 Mono, 45
 Monte Carlo, 449
 MouseDown, 137
 MouseMove, 124, 613
 MSDN, 621
 MSIL, 46, 252, 266
 MSMQ, 519, 520, 553
 MsmqIntegrationBinding, 527
 MTAThread, 72
 MultiExtended, 142
 multimedia, 671
 Multiplicity, 407
 MultiSimple, 142
 multithreading, 447
 muteksy, 466
 Mutex, 467
 mysz, 168, 727

N

nadawca zdarzenia, 33
 namespace, 25
 natywne biblioteki DLL, 616
 Navigate, 665
 nazwa okna, 103
 nazwy, 25
 negatyw obrazu, 763
 net start, 643
 net stop, 643
 NetMsmqBinding, 527
 NetNamedPipeBinding, 526
 NetPeerTcpBinding, 526
 NetTcpBinding, 526, 527
 NetTcpContextBinding, 526
 NetworkStream, 342
 new, 27, 50, 72, 80
 New Project, 20
 new[], 80
 NewArrayExpression, 362
 NewExpression, 362
 nieobsłużone wyjątki, 69
 Northwind, 207
 Notatnik.NET, 101
 NotifyFilter, 175
 NotifyIcon, 122
 null, 62, 169
 Nullable, 62
 NumberFormat, 694
 NUnit, 789

O

O/R Designer, 318, 322, 323
 obiekty, 61, 62, 83
 deserializacja, 343
 inicjacja, 80
 metody, 55, 83
 pola, 83
 serializacja, 341
 tworzenie, 72
 usuwanie z pamięci, 27
 własności, 83
 obiekty COM, 65
 obiekty POCO, 401, 441
 obiekty pozorne, 794
 object, 63, 65
 Object, 51, 590
 Object Explorer, 780
 Object Linking and Embedding, 650
 Object Relational Designer, 322
 Object Relational Mapping, 401
 Object.GetType, 587
 ObjectContext, 409, 410
 ObjectQuery<T>, 409
 ObjectServices, 409
 ObjectSet<T>, 409
 obliczanie liczby π
 metoda Monte Carlo, 492
 użycie agregacji, 509
 obliczanie odległości w przestrzeni euklidesowej, 508
 obramowanie kontrolki, 182
 obraz tła, 120
 obrazy, 148
 obserwowanie wartości zmiennych, 39
 obsługa klawiszy, 33
 obsługa kolizji, 723
 obsługa kontrolerów gry, 727
 obsługa wyjątków PLINQ, 517
 obsługa zdarzeń serwera automatyzacji, 658
 Obsolete, 72
 obszar powiadomień, 122
 odbieranie komunikatów Windows, 612
 odczytywanie
 błędy generowane w trakcie pracy usługi Windows, 645
 dane z klawiatury, 769
 dane z nieznanych źródeł, 49
 dane z rejestru Windows, 571
 pliki XML, 338
 pliki XML o nieznannej strukturze, 339
 pliki tekstowe, 109, 370
 poziom zaufania bieżącej aplikacji, 271
 położenie formy w prywatnym kluczu aplikacji, 573
 rozmiar formy w prywatnym kluczu aplikacji, 573
 ustawienia aplikacji, 127
 wersja systemu Windows, 623, 631
 zmiennie środowiskowe, 773
 odszukiwanie powtarzających się wyrazów, 787
 odświeczacz, 27
 odświeżanie komponentu, 173
 odtwarzanie multimediów, 671
 odtwarzanie pliku dźwiękowego, 125
 odwołanie do elementu tablicy, 50

odzwierciedlanie typów, 583, 587
 ograniczanie ilości instancji aplikacji, 467
 ograniczanie uprawnień, 271
 okna dialogowe, 106, 771
 ColorDialog, 112
 FontDialog, 112
 okna dialogowe serwera automatyzacji, 655
 OpenFileDialog, 110
 PageSetupDialog, 116
 PrintPreviewDialog, 116
 SaveFileDialog, 111
 ustawienia wydruku, 116
 wybór czcionki, 112
 wybór koloru, 112
 wybór pliku, 110
 okno, 21, 103
 okno o dowolnym kształcie, 147
 określanie typu zmiennej przy inicjacji, 49
 OLE, 285, 286, 650
 automatyzacja, 650
 OLE container, 650
 OLE DataBase, 650
 OLE DB, 650
 OnClosing, 673
 onColumnChanged, 218, 219
 OnContinue, 637, 640, 641
 OnMouseDown, 168
 OnPause, 637, 640, 641
 onRowChanged, 218, 219
 onRowChanging, 218, 219
 OnShutDown, 637
 OnStart, 637, 640, 641
 OnStop, 637, 640, 641
 OnValueChanged, 187
 Opacity, 150
 open source, 45
 OpenFileDialog, 110, 250, 655, 662, 677
 OpenGL, 755
 OpenSubKey, 573, 575
 OpenXML, 255
 operacje na bitach, 50
 operacje na wartościach logicznych, 50
 OperationCanceledException, 475, 477, 515, 517
 operatory, 49, 193
 ++, 50
 =>, 60
 arytmetyczne, 88
 as, 50, 51
 checked, 50
 delete, 62
 is, 50, 51
 jednoargumentowe, 50
 konwersja typów podstawowych, 51
 new, 27, 50, 72, 80
 operacje na bitach, 50
 operacje na wartościach logicznych, 50
 operator warunkowy, 50
 operatory konwersji, 90
 podstawowe operatory, 50
 porównania, 89
 przeciążanie, 88
 przypisanie, 47, 50

sizeof, 48
 typeof, 50
 unchecked, 50
 opis biblioteki, 177
 oprawa dźwiękowa, 739
 ORDER BY, 417
 orderby, 316, 351, 417
 OrderedParallelQuery<>, 512
 ORM, 401
 OSVERSIONINFO, 629, 630
 out, 59, 78
 overload, 57
 override, 86, 173

P

P/Invoke, 617
 P2P, 520
 PadLeft, 52
 PageSetupDialog, 116
 Paint, 129, 148, 182
 PaintEventArgs, 116
 Panel, 22, 27
 panel zadań aplikacji MS Excel, 690
 panele, 22
 Parallel, 447, 469, 487, 513
 For, 487, 488, 493
 ForEach, 489
 Invoke, 489
 Parallel Extensions, 469, 499
 Parallel Framework Extensions, 486
 Parallel LINQ, 447, 469
 ParallelEnumerable, 507
 ParallelExecutionMode, 514
 ParallelLoopResult, 491
 ParallelLoopState, 488, 491
 IsExceptional, 491
 IsStopped, 491
 LowestBreakIteration, 491
 ShouldExitCurrentIteration, 491
 Stop, 491
 ParallelMergeOptions, 513
 ParallelOptions, 488, 490, 513
 CancellationToken, 490
 MaxDegreeOfParallelism, 490
 TaskScheduler, 490
 ParallelQuery, 511, 512
 ParallelQuery<>, 418, 507, 512
 ForAll, 518
 ParameterExpression, 362
 ParameterizedThreadStart, 461
 parametry, 57
 parametry formalne, 57
 parametry typów ogólnych, 94
 parametryzacja kwerend, 221
 params, 78
 partial, 24
 Partitioner, 499
 pasek postępu, 181, 607, 675
 pasek stanu, 104
 paski przewijania, 102

- Path.GetFileName, 160
- PDF, 249, 297
- pętla gry, 709
- pętla, 67
 - break, 67
 - continue, 67
 - do..while, 67
 - for, 67
 - foreach, 68, 74
 - pętle nieskończone, 67
 - przejście do kolejnej iteracji, 67
 - przerwanie działania, 67
 - while, 67
 - zrównoleganie pętli, 487
- piaskownica, 272
- PictureBox, 690
- piksele, 131
- Plnvoke, 603, 610, 617
 - argumenty funkcji, 605
 - funkcje bez argumentów, 603
 - projekt, 633
 - zwracanie tablicy znaków w funkcjach WinAPI, 608
 - zwracanie wartości przez argumenty, 606
- pixel shader, 755
- Plain Old CLR Object, 441
- planista zadań, 482
 - zarządzanie zadaniami, 485
- Platform Invoke, 603, 617
- platforma .NET, 15, 45, 46
- platforma Mono, 45
- platforma obiektów pozomych, 794
- PlayerSprite, 733
- Plik, 114
- pliki, 159
 - edmx, 405
 - exe, 46
 - ico, 104
 - odczytywanie, 109
 - SVC, 539
 - ścieżka dostępu, 160
 - wav, 125, 739
 - XML, 250, 254
- pliki bazy danych, 312
- pliki tekstowe, 106, 179, 367
 - LINQ to TXT, 367
 - odczytywanie, 109
 - zapisywanie, 111
- pliki XML, 335
 - ADO.NET, 343
 - deserializacja obiektu, 343
 - LINQ to XML, 346
 - modyfikacja, 351
 - odczytywanie, 338
 - odczytywanie atrybutów elementów, 341
 - odczytywanie pliku o nieznanym strukturze, 339
 - serializacja obiektów, 341
 - tworzenie, 346
 - wczytywanie danych do komponentu DataSet, 344
 - XmlReaderSettings, 340
 - XmlSerializer, 342
 - XmlTextReader, 336, 339
 - XmlTextWriter, 336
 - zapisywanie, 337
- PLINQ, 447, 469, 486, 487, 511
 - algorytm podziału danych, 513
 - CancellationToken, 515
 - CancellationTokenSource, 515
 - ForAll, 517
 - funkcje haszujące, 513
 - interfejsy, 512
 - metody przekształcające dane wynikowe, 515
 - obsługa wyjątków, 517
 - ParallelExecutionMode, 514
 - ParallelMergeOptions, 513
 - podział na „paski”, 513
 - podział zakresu, 513
 - przerywanie zapytań, 515
 - scalanie danych, 513
 - segmentacja, 513
 - wyjątki, 517
 - zapytania, 511
 - zastosowanie, 514
- pobieranie danych, 306, 319
- pobieranie danych ze źródeł, 303
- pobieranie zawartości katalogu, 158
- Pocket PC, 45
- POCO, 401, 441
- podgląd wydruku, 116
- Podgląd zdarzeń, 645
- podglądanie wartości zmiennych, 40
- podraporty, 295
- podwójna kompilacja, 46
- podwójne buforowanie, 131
- podzespóły, 569
- podział linii uwzględniający wyrazy, 118
- podział na „paski”, 513
- podział zakresu, 513
- pojemnik OLE, 650
- pokazywanie paska stanu, 108
- pola, 83, 205
- pola statyczne, 85
- polecenia SQL, 332
- połączenia P2P, 520
- połączenie raportu z aplikacją, 277
- połączenie z bazą danych, 265
- połączenie ze źródłem danych, 206, 207
- Pong, 713
- pop, 501
- porównania, 89
- port szeregowy, 155
- PostMessage, 609
- pośrednik, 272
- potok renderowania, 755
- potoki, 520
- potwierdzenie zamknięcia aplikacji, 107
- powtarzające się wyrazy, 787
- poziom izolacji transakcji, 264
- poziom zaufania bieżącej aplikacji, 271
- predefiniowane kolory, 30
- PreferFairness, 483, 485
- preprocesor, 70
- prezentacja danych, 233
- prezentacja danych w postaci macierzowej, 287
- PrintDialog, 114, 115
- PrintDocument, 114

- PrintOut, 663
- PrintPage, 114, 115
- PrintPageEventArgs, 116
- PrintPreview, 663
- PrintPreviewControl, 117
- PrintPreviewDialog, 116
- priorytety wątku, 455
- private, 84, 151
- procedury składowane, 331
 - Entity Framework, 424
 - modyfikacja danych, 331
 - pobieranie danych, 331
 - wykonywanie poleceń SQL, 332
- Process, 133
- Process Explorer, 627
- Process.GetProcess, 133
- procesy, 132
- profil użytkownika, 771
- programowanie równoległe, 447
- programowanie współbieżne, 447, 469
 - BlockingCollection, 501
 - CollectionBag, 500
 - ConcurrentQueue, 501
 - ConcurrentStack, 501
 - dostęp do źródła danych, 499
 - Interlocked, 499
 - IProducerConsumerCollection, 499
 - kolejka, 501
 - kolekcje, 500
 - obliczanie liczby π metodą Monte Carlo, 449
 - Parallel, 487
 - Parallel Extensions, 499
 - Partitioner, 499
 - PLINQ, 511
 - stos, 501
 - Task, 469
 - Thread, 449
 - tworzenie kolekcji współbieżnych, 503
 - wątki, 449
 - współbieżne struktury danych, 499
 - zadania, 469
 - zamrożenie wątku, 499
 - zbiory danych, 499
 - zrównoleglanie pętli, 487
- ProgressBar, 155, 181, 607, 675
- projekt, 20
 - aplikacje konsolowe, 767
 - aplikacje okienkowe, 20
 - biblioteki DLL, 584, 632
 - Class Library, 667
 - dodatki do skoroszytu, 682
 - instalator usługi Windows, 644
 - PIInvoke, 633
 - serwer automatyzacji, 667
 - tworzenie, 20
 - ustawienia, 103
 - WCF, 529
 - wczytywanie, 24
 - Windows Service, 637
 - XACT, 740
 - zapisywanie, 24
- projektant obiektowo-relacyjny, 322
- projektowanie
 - aplikacje, 19
 - aplikacje bazodanowe, 275
 - graficzny układ raportu, 283
 - interfejs aplikacji, 20, 101
 - interfejs kontrolki, 157
 - klasa encji, 322
 - kontrolki, 155
 - projektowanie wizualne, 19, 88
 - projektowanie zorientowane obiektowo, 83
 - raporty, 279
 - serwer automatyzacji, 667
- Properties, 24, 128
- Properties Window, 21
- protokoły
 - HTTP, 266, 525
 - SSL, 266
 - TCP, 526
- provider connection string, 402
- próbkowanie tekstur, 756
- przebiegi, 758
- przechwytywanie wyjątków, 42, 68
 - zadania, 477
- przeciąganie, 137
- przeciągnij i upuść, 135
 - akceptacja, 135
 - akceptacja upuszczenia elementu, 138
 - czynności wykonywane po zakończeniu procesu przenoszenia i upuszczania, 140
 - DoDragDrop, 135
 - DragDrop, 135, 138
 - DragOver, 135, 138, 139
 - inicjacja procesu przeciągania, 137
 - MouseDown, 137
 - opóźnione inicjowanie procesu przenoszenia, 141
 - przenoszenie elementów między różnymi aplikacjami, 140
 - przenoszenie plików, 144
 - przenoszenie wielu elementów, 142
 - reakcja na upuszczenie elementu, 135, 139
 - rozpoczęcia przenoszenia, 135
- przeciążanie metod, 56
- przeciążanie operatorów, 88
- przeglądanie plików tekstowych, 179
- przeglądarka rozwiązania, 25
- przeglądarka skojarzeń plików, 575
- przekazywanie argumentów
 - ciągi znakowe, 634
 - przekazywanie przez referencje, 59
 - przekazywanie przez wartości, 58
 - przekazywanie struktury do funkcji niezarządzanych, 628
 - wskaźniki, 634
- przenoszenie danych z bazy danych (komponentu DataSet) do pliku XML, 350
- przenoszenie danych z kolekcji do pliku XML, 349
- przenoszenie formy za dowolny punkt, 151
- przerywanie
 - pętle, 490
 - wątki, 453
 - zadania, 475
 - zapytań PLINQ, 515

przeźrenie nazw, 25
 przesunięcie bitów, 50
 przesyłanie danych do wątku, 461
 przetwarzanie pikseli, 755
 przetwarzanie tablic, 74
 przewijanie odtwarzanego pliku, 675
 przezroczyste okna o dowolnym kształcie, 147, 150
 przypisanie, 47, 50
 przypisywanie metody do zdarzeń komponentów, 31
 public, 84, 85
 pudełkowanie, 63
 pula wątków, 463
 liczba wątków, 464
 ThreadPool, 464
 tworzenie, 463
 Pulse, 465
 punkt przerwania, 38
 punkt wejściowy aplikacji, 21, 768
 push, 501
 Pytanie – odpowiedź, 525
 Python, 46

Q

Query, 359, 360
 QueryInterface, 649
 Queue, 77
 QueueTask, 483
 Quit, 654

R

Random, 495
 RandomThreadSafe, 496, 500
 Range, 253, 664
 range partitioning, 513
 RanToCompletion, 477, 478
 raportowanie, 277
 Chart, 294
 dynamiczne filtrowanie danych, 290
 eksport danych, 297
 etykiety, 284
 filtr definiowany na poziomie raportu, 289
 filtrowanie danych, 289
 filtrowanie raportu na etapie pobierania danych
 źródłowych, 290
 format prezentacji danych, 281
 Gauge, 292
 grupowanie danych, 282
 kluczowe wskaźniki efektywności, 292
 kolumny obliczeniowe raportu, 280
 konfiguracja połączenia ze źródłem danych, 278
 konfiguracja źródła danych, 278
 listy, 283, 284
 nagłówki, 286
 OLE, 286
 parametry raportu, 289
 podraporty, 283, 295
 prezentacja danych w postaci macierzowej, 287
 projektowanie graficznego układu raportu, 283
 projektowanie raportu, 279

raporty drill-through, 295
 raporty wielowarstwowe, 295
 RefreshReport, 284
 ReportViewer, 277
 rysunki, 284
 stopka, 286
 tworzenie raportu, 277
 wiązanie danych, 286
 wskaźniki graficzne, 283, 292
 wykresy, 283, 294
 raporty, 277
 raporty wielowarstwowe, 295
 rasteryzacja, 755
 raw data, 367
 RCW, 649
 RDL, 301
 rdzenie procesora, 456
 Read, 769
 ReadElementString, 339
 ReadEndElement, 339
 reader-writer lock, 466
 ReadLine, 769
 ReadOnly, 577
 ReadStartElement, 339
 ReadToEnd, 109
 reakcja na naciskanie klawiszy, 33
 ref, 59, 78, 608
 Refactor, 628
 referencja do biblioteki systemowej platformy .NET, 584
 referencje, 59, 62
 reflection, 65, 583, 587, 595
 Reflection, 355
 Refresh, 173, 185
 RefreshReport, 284
 regedit, 577
 Regex, 783
 IsMatch, 784
 Replace, 787, 788
 Split, 787
 Register for COM Interop, 668
 Registry, 571, 578
 RegistryKey, 573, 579
 RegularExpressions, 781
 reguły integralności danych, 225, 227
 RejectChanges, 216
 rejestr systemu Windows, 571
 informacje o typach plików, 575
 klucze, 573
 klucze główne, 573
 odczytywanie danych z rejestru, 571
 odczytywanie listy rozszerzeń plików, 578
 otwieranie klucza, 572
 prywatne klucze aplikacji, 573
 Registry, 571, 578
 RegistryKey, 573
 skojarzenia plików, 575
 usuwanie klucza, 575
 zamykanie klucza, 572
 rekordy, 205
 relacje, 205, 404
 relacyjna baza danych, 205
 Release, 649

Remove, 52, 53, 54
 renderowanie grafiki, 755
 Replace, 52, 53, 54
 Report Definition Language, 301
 ReportViewer, 277, 278, 297
 ReportViewer.LocalReport, 301
 Request – reply, 525
 Resume, 452, 454
 return, 58
 Rhino Mocks, 794
 Ribbon, 687
 RichTextBox, 664
 rollback transaction, 257
 ROT, 252, 651
 Routing Service, 568
 Rows.Autofit, 695
 rozmiar tablicy, 73
 rozszerzenia, 98
 rozszerzenia na poziomie dokumentu, 681
 równoległa pętla for, 487
 równoległe obliczanie odległości w przestrzeni euklidesowej, 508
 równoległe wykonanie dużej ilości zadań, 470
 równoległe zapytania LINQ to Entities, 418
 RSACryptoProvider, 268
 Ruby, 46
 Run Tests, 792
 Run to Cursor, 38
 Running, 478
 Running Object Table, 252, 651
 Runtime Callable Wrapper, 649
 RunWorkerAsync, 119
 rysowanie, 129

- buforowanie, 131
- fraktale, 198
- obramowanie kontrolki, 182
- obraz, 714
- pasek postępu, 185
- wzór, 130
- zbiór Mandelbrota, 200

 rysunki, 284
 rzutowanie, 50

S

sampler, 756
 SandBox, 272
 Save All, 24
 SaveFileDialog, 111, 300
 sbyte, 48
 SByte, 48
 SC_SCREENSAVE, 611
 schemat XML, 352
 schowek, 113
 schowek systemowy, 250
 ScrollBars, 102
 SectionInformation.IsProtected, 269
 Secure Socket Layer, 266
 SecurityException, 270, 272
 segmentacja, 513
 sekcje krytyczne, 460
 select, 310, 351, 412
 SELECT, 411
 SelectedIndexChanged, 168
 SelectedValueChanged, 233, 234, 290
 Selection, 664
 SelectionMode, 142
 Self-Hosting, 534, 535
 semafony, 466
 semantyki wejściowe shaderów pikseli, 757
 semantyki wejściowe shaderów wierzchołków, 757
 semantyki wyjściowe shaderów pikseli, 758
 semantyki wyjściowe shaderów wierzchołków, 757
 SemaphoreSlim, 499
 sender, 33
 SendMessage, 609, 611
 serializacja obiektów do pliku XML, 341
 Serialize, 342
 SerialPort, 155
 Service Oriented Architecture, 520
 ServiceController, 646, 648
 ServiceInstaller, 641
 ServiceProcessInstaller, 641
 serwer automatyzacji, 650, 667

- MS Excel, 650
- MS Word, 661

 serwer IIS, 538
 serwis routujący, 562
 serwis windowsowy, 540
 SetLastError, 770
 SetIn, 770
 SetOut, 770
 SetPixel, 131
 Settings, 128
 shadery, 755, 759
 shadery pikseli, 755
 shadery wierzchołków, 755
 shdocvw.dll, 665
 SheetChange, 658
 SheetSelectionChange, 658
 ShellExecute, 606
 short, 48
 ShortcutDisplayString, 152
 ShortcutKeys, 106
 ShouldExitCurrentIteration, 491
 ShowBalloonTip, 125
 ShowDialog, 110
 ShowInTaskbar, 148
 Silverlight, 45
 Simple Object Access Protocol, 520
 simplex, 525
 Single, 48, 307
 Size, 27
 sizeof, 48
 skojarzenia plików, 575
 Sleep, 453
 słowa kluczowe

- break, 67
- case, 66
- catch, 42, 68
- class, 25, 61, 84
- continue, 67
- default, 48

- delegate, 59, 60, 98
- do, 67
- dynamic, 63, 595
- else, 66
- event, 59
- explicit, 91
- finally, 69
- for, 67
- foreach, 74
- if, 66
- implicit, 91
- lock, 461
- namespace, 25
- null, 62
- out, 59
- override, 86
- partial, 24
- ref, 59, 608
- return, 58
- static, 85
- struct, 61, 84
- switch, 66
- this, 27, 98
- throw, 41, 70
- try, 42, 68
- using, 25
- var, 49
- void, 56
- while, 67
- yield, 79
- słowniki, 77
- SM_CXDRAG, 141
- SM_CYDRAG, 141
- SOA, 520, 568
- SOAP, 520
- Solution, 24
- Solution Explorer, 25, 208
- SortedDictionary, 77
- SortedList, 77
- SortedList<>, 72
- sortowanie, 74, 92
- sound banks, 739
- SoundBank, 742
- SoundPlayer, 120, 125
- SpecialFolders, 571
- SpinLock, 466, 499
- SpinWait, 471, 499, 514
- splash screen, 120
- SplitContainer, 135, 179
- SplitterDistance, 136
- społeczność open source, 45
- sprite, 708
- Sprite, 714, 717, 722
- SpriteBatch, 708, 711, 714, 733, 748
- SpriteSortMode.Immediate, 761
- SQL, 303, 311, 317
- SQL Injection, 275
- SQL Server, 206, 257, 311
- SQL Server 2008 Express Edition, 775
 - Centrum Instalacji SQL Server, 775
 - instalacja, 775
 - wybór trybu SQL Server, 778
- SQL Server Management Studio, 779
- SQL Server Reporting Services, 277
- SqlCommand, 401
- SqlConnectionCAS, 271, 272
- SqlConnection, 260
 - BeginTransaction, 260
- SqlDataReader, 401
- SqlTransaction, 258, 260
 - Commit, 260
 - Rollback, 260
- SSL, 266
- SSRS, 277
- Stack, 77
- stałe liczbowe, 49
- stałe procesora, 70
- stan komponentu, 157
- stan wyjątkowy, 41
- stan zadania, 478
- Start, 452
- StartFromScratch, 690
- StartNew, 481
- StartPosition, 121, 136
- StartsWith, 52
- static, 85, 604
- StatusStrip, 104, 647
- statyczne ładowanie biblioteki DLL, 583, 586
- statyczne obiekty składowe, 84, 85
- step into, 37
- step out, 37
- Step Over, 37
- sterowanie przepływem, 66
- Stop Debugging, 122
- stored procedures, 331
- stos, 77
- stos wywołań, 41
- StreamReader, 109, 179
- StreamWriter, 342
- string, 48, 52, 53
- String, 52, 53
- StringBuilder, 53, 54, 608, 634
- StringReader, 115
- strip partitioning, 513
- struct, 61, 84, 190
- struktury, 61, 84, 85, 190
 - elementy składowe, 84
 - konstruktor, 84, 85
 - metody statyczne, 85
 - pola statyczne, 85
- struktury danych, 77
- SubmitChanges, 331
- Substring, 52
- Sum, 307
- Suspend, 452, 454
- suwaki, 22
- SVC, 539
- SvcUtil.exe, 547, 548
- switch, 66
 - case, 66
- sygnały dźwiękowe, 120
- synchronizacja pętli równoległych, 492
- synchronizacja wątków, 460, 466
- synchronizacja zadań, 473

synchronizacja zmiennej, 494
 SyncRoot, 504
 System, 767
 system zarządzania bazą danych, 205
 System.ArgumentOutOfRangeException, 76
 System.Array, 54, 72, 73
 System.Byte, 54
 System.Collections, 72, 354
 System.Collections.Concurrent, 499, 500, 501
 System.Collections.Generic, 72, 77, 767
 System.Collections.Generic, 77
 System.Collections.Generic.List, 75
 System.Collections.Specialized, 72
 System.Delegate, 54
 System.Diagnostics, 132
 System.Diagnostics.Process.GetProcesses, 610
 System.Diagnostics.Process.Start, 606
 System.DllNotFoundException, 619
 System.Drawing.Bitmap, 149
 System.EnterpriseServices, 519
 System.Enum, 54
 System.Environment, 571
 System.Environment.ProcessorCount, 456
 System.Environment.SystemDirectory, 608
 System.Exception, 41
 System.Int32, 48, 49
 System.Int64, 49
 System.InvalidOperationException, 75
 System.IO, 158, 179, 342
 System.IO.DriveInfo, 606
 System.IO.Pipes, 520
 System.Lazy, 601, 602
 System.Linq, 767
 System.Linq.Enumerable, 360
 System.Media, 120
 System.Media.SystemSounds.Beep.Play, 603
 System.Messaging, 609
 System.Net.PeerToPeer, 520
 System.Net.Sockets, 520
 System.NullReferenceException, 73
 System.Runtime.InteropServices, 610, 617, 621
 System.Runtime.Remoting, 520
 System.Runtime.Serialization.dll, 524
 System.ServiceModel, 523, 536, 563
 System.ServiceModel.Configuration, 523
 System.ServiceModel.Description, 523
 System.ServiceModel.Discovery, 559
 System.ServiceModel.dll, 523
 System.ServiceModel.MsmqIntegration, 523
 System.ServiceModel.Security, 523
 System.ServiceProcess.ServiceController, 646
 System.String, 52
 System.Text, 767
 System.Text.RegularExpressions, 781
 System.Threading, 120, 469, 490
 System.Threading.Tasks, 469
 System.Type.Missing, 253
 System.Value, 54
 System.ValueType, 191
 System.Windows.Forms, 771
 System.Windows.Forms.Form, 107
 System.Windows.Forms.MessageBox, 612

System.Windows.Forms.MessageBox, 585
 System.Xml.Linq, 346
 System.Xml.Serialization, 342
 SystemColor, 30, 101
 SystemDirectory, 608
 SystemInfo, 585, 586, 595
 SystemSounds, 603
 szablon projektu dodatku do skoroszytu, 682
 sztafeta zadań, 474
 szyfrowane połączenie ze źródłem danych, 265
 szyfrowanie danych, 268
 szyfrowanie połączenia, 265

Ś

ścieżka dostępu do plików, 160
 śledzenie wykonywania pętli, 39
 śledzenie wykonywania programu krok po kroku, 37
 środowisko aplikacji, 771
 środowisko uruchomieniowe, 46, 616
 środowisko Visual Studio, 19

T

tabele, 205
 TabIndex, 27
 TableAdapter, 206, 221, 239, 275
 edycja danych, 224
 integralność danych, 224
 usuwanie danych, 224
 wstawianie danych, 228
 tablica ROT, 252, 651
 tablice, 72
 argumenty metod, 78
 deklaracja, 72
 inicjacja, 80
 przetwarzanie, 74
 rozmiar, 73
 sortowanie, 74
 tablice dynamiczne, 72
 tablice wielowymiarowe, 73
 Target Framework, 104
 Task, 447, 469
 ContinueWith, 474
 Start, 470
 Wait, 473
 WaitAll, 473
 Task Parallel Library, 447, 469
 TaskContinuationOptions, 475, 482
 TaskCreationOptions, 480, 482, 483, 485
 TaskFactory, 469, 480
 ContinueWhenAll, 481
 ContinueWhenAny, 481
 StartNew, 480, 481
 TaskScheduler, 469, 482, 483, 485, 490
 TaskStatus, 478
 TCP, 526
 technologia .NET, 45
 technologia COM, 649
 technologia MEF, 595
 technologie Windows, 569

- tekst podpowiedzi ikony w zasobniku, 124
- teoria Murphy'ego, 35
- Test Driven Development, 789
- test liczby pierwszej, 472
- testowanie operatorów arytmetycznych, 89
- testy jednostkowe, 789
 - etapy, 789
 - metody czyszczące, 794
 - metody inicjujące, 794
 - obiekty pozorne, 794
 - platformy testowe, 789
 - przygotowanie testów, 790
 - sprawdzanie wyniku testu, 792
 - tworzenie testów, 791
- tex2D, 761
- TextBox, 108, 109, 577
 - Lines, 109
- TextChanged, 108
- TextRender, 113
- this, 27, 33, 98
- Thread, 120, 447, 449, 451
 - Abort, 453
 - Join, 458, 465
 - Resume, 454
 - Sleep, 453, 471
 - SpinWait, 471, 514
 - Start, 452
 - Suspend, 454
 - Wait, 470
- ThreadPool, 464, 482, 513
 - GetAvailableThreads, 464
 - GetMaxThreads, 464
- ThreadPriority, 455
- ThreadStart, 451, 461
- ThreadState, 452
- ThreadStatic, 496
- throw, 41, 70
- ThrowIfCancellationRequested, 516
- Tick, 126
- TickCount, 772
- Timer, 122, 125, 155, 639, 641
- timery, 122
- TimeSpan, 263, 710
- ToArray, 504
- ToDouble, 86
- ToLower, 52
- Toolbox, 21, 22, 23
 - dokowanie, 22
 - umieszczenie kontrolki, 177
- ToolStripStatusLabel, 105
- ToolTip, 233, 234, 242, 278
- ToString, 48, 86, 88, 192
- ToUpper, 52
- TPL, 447, 469
- TrackBar, 22, 27, 675
- TransactionScope, 261, 262
 - Complete, 262
- TransactionScopeOption, 262
- transakcje, 257
 - ADO.NET, 261
 - automatyczne zarządzanie transakcjami, 261
 - konkurencyjność, 264
 - kontrola, 257
 - poziom izolacji, 264
 - SqlConnection, 260
 - TransactionScope, 261, 262
 - TransactionScopeOption, 262
 - tworzenie, 257
 - wycofanie, 257
- transfer danych do plików Excela, 249
- transformacja klasy w strukturę, 84
- transformacja wierzchołków, 755
- TransparentKey, 147
- TreeView, 654
- trigger, 187
- Trim, 52
- TrimEnd, 52
- TrimStart, 52
- TrustServerCertificate, 266
- try..catch, 41, 42, 68, 624
- TryAdd, 499, 504
- TryExecuteTaskInline, 483, 485
- TryTake, 504
- T-SQL, 257, 275, 317, 428
- tworzenie
 - aplikacje Web, 538
 - dodatki dla pakietu Office, 682
 - fabryka zadań, 482
 - gry XNA, 707
 - kolekcje współbieżne, 503
 - komponenty, 156
 - metody zdarzeniowe, 29
 - model encji, 432
 - obiekty, 72
 - panel zadań aplikacji MS Excel, 690
 - planista zadań, 483
 - pliki XML, 346
 - projekt, 20, 102
 - projekt XACT, 740
 - przestrzenie nazw, 25
 - pula wątków, 463
 - raporty, 277
 - referencja do serwera automatyzacji, 669
 - serwer automatyzacji, 667
 - transakcje, 257
 - usługi Windows, 638
 - ustawienia aplikacji, 126
 - wątki, 120, 451
 - Windows Service, 540
 - wykresy w skrószycie, 686
 - wyrażenia regularne, 783
 - zadania, 469
 - zadania przez fabrykę obiektów, 481
 - zapytania równoległe LINQ To Entities, 418
 - zarządzane biblioteki DLL, 584
 - źródła danych LINQ, 353
- type library, 650
- typedef, 635
- typeof, 50, 359
- typy anonimowe, 99
- typy danych, 47
 - bool, 48
 - boxing, 63
 - byte, 48

typy danych
 char, 48, 53
 decimal, 48
 double, 48
 dynamic, 595, 682
 float, 48
 int, 48
 liczby, 48
 long, 48
 pudełkowanie, 63
 rozmiar, 48
 sbyte, 48
 short, 48
 string, 48
 typy referencyjne, 61
 typy wartościowe, 61
 uint, 48
 ulong, 48
 unboxing, 63
 ushort, 48
 wartość domyślna, 48
 wyliczenia, 54
 znaki, 48
 typy dynamiczne, 63
 typy MIME, 286
 typy ogólne, 75, 92
 definiowanie, 93
 typy parametryczne, 92
 implementacja interfejsów, 95
 parametry, 94
 wiele parametrów, 97
 typy plików, 575

U

uchwyt do metody, 60
 uchwyt okna, 610, 611
 uint, 48
 UInt16, 48
 UInt32, 48
 UInt64, 48
 układ współrzędnych, 719
 ukrywanie
 okno po dwóch sekundach od uruchomienia
 aplikacji, 122
 pasek stanu, 108
 standardowe zakładki wstążki menu w aplikacji
 Excel, 690
 ulong, 48
 umieszczenie kontrolki w podoknie Toolbox, 177
 UnaryExpression, 362
 UnauthorizedAccessException, 109
 unboxing, 63
 unchecked, 50
 Unicode, 33, 52
 Union, 308
 Unit Testing Frameworks, 789
 unit tests, 789
 UnloadContent, 709
 uprawnienia, 271
 URL, 786

uruchamianie
 metody na rzecz instancji obiektu, 593
 metody statyczne z klasy wczytanej z biblioteki
 DLL, 593
 serwer automatyzacji MS Excel, 652
 SQL Server Management Studio, 779
 wątki, 451
 wygaszacz ekranu, 611
 zadania, 470
 user.config, 128
 UserControl, 157, 168
 USERPROFILE, 773
 ushort, 48
 using, 25, 771
 usługi sieciowe, 519
 SOA, 520
 WCF, 520, 523
 usługi Windows, 637
 instalacja, 637, 641
 LocalAccount, 642
 odczytywanie błędów, 645
 OnContinue, 637, 640, 641
 OnPause, 637, 640, 641
 OnShutDown, 637
 OnStart, 637, 640, 641
 OnStop, 637, 640, 641
 Podgląd zdarzeń, 645
 projekt, 637, 641
 projekt instalatora usługi, 644
 ręczna instalacja, 641, 642
 ServiceController, 648
 ServiceInstaller, 641
 ServiceProcessInstaller, 641
 stan usługi, 647
 StartType, 642
 tworzenie, 638
 uruchamianie, 640
 wstrzymanie działania, 640
 wznowienie działania, 640
 zarządzanie usługą z poziomu innej aplikacji, 646
 zatrzymywanie, 640
 ustawianie breakpointa, 39
 ustawienia aplikacji, 126
 app.config, 127
 odczytywanie ustawień, 127
 tworzenie, 126
 zapisywanie ustawień, 128
 ustawienia pętli równoległych, 490
 ustawienia projektu, 103
 Ustawienia strony..., 114
 ustawienia zadań, 485
 usuwanie
 błędy, 36
 dane z zachowaniem reguł integralności, 227
 klucze rejestru, 575
 obiekty, 27
 rekordy, 220
 usypianie wątku, 452
 uwierzytelnianie w serwerze Microsoft SQL Server, 274

V

value, 77
 Value, 77, 187
 VALUE, 411
 ValueChanged, 29, 31
 var, 49, 64
 VBA, 681
 Visible, 108, 123, 652
 Visual C# 2010 Express, 19
 Visual Studio 2010, 15, 19
 IntelliSense, 30
 klawisze skrótów edytora, 31
 okno własności, 21
 paleta komponentów, 22
 Properties Window, 21
 raportowanie, 277
 Solution Explorer, 25
 Toolbox, 21, 22
 tworzenie projektu, 20
 wczytywanie projektu, 24
 zapisywanie projektu, 24
 Visual Studio Tools for Office, 681
 void, 49, 56, 65
 VSTO, 681
 dodatki na poziomie aplikacji, 681, 696
 formatowanie komórek, 693
 implementacja funkcjonalności aplikacji, 698
 instalacja dodatku, 702
 komponenty Windows Forms, 684
 modyfikacja wstążki, 687
 obsługa zdarzeń, 695
 panel zadań aplikacji MS Excel, 690
 połączenie skoroszytu ze źródłem danych, 694
 projekt dodatku do skoroszytu, 682
 projektowanie dodatku, 696
 rozszerzenia na poziomie dokumentu, 681
 tworzenie wykresu w skoroszytcie, 686
 ukrywanie standardowych zakładek wstążki menu
 w aplikacji Excel, 690
 usuwanie dodatku, 702
 zarządzanie dodatkami w aplikacji MS Excel, 702

W

Wait, 470, 473
 WaitAll, 473, 477
 WaitCallback, 464
 WaitingForActivation, 478
 WaitingToRun, 478
 WaitOne, 465
 walidacja adresu URL, 786
 wartości, 77
 wartości zwracane przez metody, 58
 wartość null, 62
 WAS, 542
 Watch, 39
 wav, 739
 wave banks, 739
 WaveBank, 742

wątki, 28, 119, 447, 449
 Abort, 452, 453
 BackgroundWorker, 119, 451
 Barrier, 466
 czekanie na ukończenie pracy wątku, 458
 EventWaitHandle, 465
 IsBackground, 452
 Join, 458
 komunikacja między wątkami, 465
 lock, 461
 Monitor, 465
 muteksy, 466
 ParameterizedThreadStart, 461
 przerywanie działania wątku, 453
 przesyłanie danych do wątku, 461
 pula wątków, 463
 Pulse, 465
 Resume, 452, 454
 sekcje krytyczne, 460
 semafony, 466
 Sleep, 453
 SpinLock, 466
 stan, 452
 Start, 452
 Suspend, 452, 454
 sygnalizacja zakończenia wątku, 465
 synchronizacja wątków, 460, 466
 Thread, 451
 ThreadPriority, 455
 ThreadStart, 451
 ThreadState, 452
 tworzenie, 120, 451
 uruchamianie, 451
 usypianie, 452
 WaitOne, 465
 wątki tła, 455
 wstrzymywanie działania, 454
 wznawiane działania, 454
 zarządzanie wątkiem, 452
 zmiana priorytetu wątku, 455
 WCF, 519, 520, 523
 .NET 4.0, 557
 <behaviors>, 551
 <bindings>, 551
 <client>, 553
 <clientBehaviors>, 551
 <commonBehaviors>, 553
 <extensions>, 553
 <routing>, 554
 <service>, 554
 <serviceBehaviors>, 551
 <serviceHostingEnvironment>, 554
 <serviceThrottling>, 551
 <standardEndpoints>, 554
 <system.serviceModel>, 550
 adres, 528
 aplikacja hostująca, 529
 BasicHttpBinding, 526
 biblioteki, 523
 definicja filtra, 567
 definicja routowania, 566
 definicja serwisu, 529

- WCF
 - definicja serwisu routującego, 565
 - definiowanie kontraktu, 524
 - discovery, 558, 560
 - filtry, 567
 - hosting, 534
 - IGeometryService, 557
 - IIS, 538
 - implementacja kontraktu, 531
 - klient, 529, 542
 - kolejki MSMQ, 527, 553
 - komunikacja typu duplex, 525
 - komunikacja typu „pytanie – odpowiedź”, 526
 - komunikacja typu simplex, 525
 - konfiguracja, 550, 557
 - konfiguracja klienta, 545
 - konfiguracja klienta korzystającego z discovery, 561
 - konfiguracja punktu końcowego korzystającego z TCP, 537
 - konfiguracja serwera, 537
 - konfiguracja serwisu, 551
 - kontrakt, 533, 524, 530
 - kontrakt serwisu, 532
 - kontrakt usługi, 533
 - menu kontekstowe plików *.config, 547
 - MEX, 532, 536
 - narzędzia, 547
 - NetMsmqBinding, 527
 - NetTcpBinding, 527, 553
 - obiekty, 530
 - plik konfiguracyjny, 531
 - plik SVC, 539
 - programowe dodanie funkcji discovery do hosta, 560
 - projekt, 529
 - protokół HTTP, 525
 - protokół TCP, 526
 - przestrzenie nazw, 523
 - punkt końcowy serwisu, 560
 - routing, 566
 - Routing Service, 568
 - schemat rozdzielania wiadomości, 567
 - Self-Hosting, 534, 535
 - serwis, 529
 - serwis bazujący na uproszczonej konfiguracji, 558
 - serwis routujący, 562
 - serwis windowsowy, 540
 - SOA, 568
 - SvcUtil.exe, 547, 548
 - udostępnianie usług, 542
 - uproszczona konfiguracja, 557
 - ustawienia wiązań, 552
 - WAS, 542
 - WcfTestClient.exe, 549
 - wiązania oparte o HTTP, 552
 - wiązanie, 524, 551
- WCF Service Application, 530, 539
- WCF Service Host, 532
- WCF Service Library, 529, 530
- WcfTestClient.exe, 549
- wczytywanie
 - dane z pliku XML do komponentu DataSet, 344
 - obraz, 148
 - obraz tekstury, 717
 - plik tekstowy, 179
 - projekt, 24
 - ustawienia z pliku XML, 338
- Web Forms, 277
- WebBrowser, 665
- WebConfigurationManager, 270
- WebHttpBinding, 525
- wektory, 756
- wersja systemu Windows, 623, 631
- wersje funkcji WinAPI, 627
- wersjonowanie obiektu DataRow, 216
- weryfikacja liczby pierwszej, 472
- weryfikacja obecności klasy o znanej nazwie w bibliotece DLL, 589
- weryfikacja obecności metody w klasie z biblioteki DLL, 590
- where, 316, 351
- while, 67
- wiązanie, 551
 - wiązania oparte na MSMQ, 527
 - wiązania oparte o protokół HTTP, 525
 - wiązania oparte o protokół TCP, 526
- wiązanie danych z kontrolkami, 233
 - BindingContext, 234
 - Chart, 237
 - DataBinding, 234
- wiązanie zdarzeń kontrolki ze zdarzeniami listy będącej jej elementem, 168
- Width, 129
- wiele do wielu, 404
- wielkość liter, 25
- wielowątkowość, 447
 - generator liczb pseudolosowych, 496
 - klasa Random, 496
- wierzchołki, 755
- więzy integralności, 226
- Win32, 569
- WinAPI, 569, 603, 615
- Windows, 569
- Windows Activation Services, 542
- Windows Communication Foundation, 520
- Windows Data Protection API, 268
- Windows Embedded, 45
- Windows Forms, 22, 46, 101, 147, 155
 - wiązanie danych, 233
- Windows Forms Application, 20, 55, 305
- Windows Forms Control Library, 156, 175
- Windows Media Player, 671, 672
 - axWindowsMediaPlayer.fullscreen, 676
 - implementacja listy odtwarzanych plików, 676
 - kontrola głośności odtwarzania, 675
 - lista odtwarzanych plików, 676
 - odtwarzanie plików multimedialnych, 671
 - prezentowanie informacji o postępie odtwarzania pliku, 672
 - przewijanie odtwarzanego pliku, 675
 - przyciski sterujące odtwarzaniem, 673
 - settings, 675
 - widok pełnego ekranu, 676
 - wstrzymywanie odtwarzania pliku, 672
 - wybór odtwarzanego pliku, 674
 - wyciszenie odtwarzanego pliku, 675
 - wznawianie odtwarzania pliku, 672

- Windows Mobile, 45
 - Windows Phone, 45
 - Windows Presentation Foundation, 101
 - Windows Service, 637
 - WinExec, 605
 - WinUser.h, 627
 - WithDegreeOfParallelism, 513, 518
 - WithExecutionMode, 518
 - WithMergeOptions, 518
 - wizualne projektowanie klasy encji, 322
 - wizytówka, 120
 - właściwości, 83, 87, 164
 - Anchor, 24
 - Dock, 24
 - komponenty, 32
 - właściwości tylko do odczytu, 166
 - WM_CLOSE, 611
 - WM_CTLCOLORLISTBOX, 612
 - WM_NCMOUSEMOVE, 613, 614
 - WM_SYSCOMMAND, 611
 - WMPLib.IWMPPlaylist, 680
 - WndProc, 609, 612
 - Word, 661
 - Word.Application, 661
 - work stealing, 483
 - Workbook.SaveAs, 256
 - Workbooks.OpenXML, 255
 - Worksheets.Count, 655
 - WPF, 101
 - WriteComment, 338
 - WriteElementString, 338
 - WriteEndElement, 338
 - WriteLine, 111, 768, 769, 770
 - WriteStartDocument, 338
 - WriteStartElement, 338
 - WriteXML, 255, 256
 - WS-Discovery, 558
 - WSDualHttpBinding, 525
 - WSFederationHttpBinding, 525
 - WSHttpBinding, 524, 525
 - WSHttpContextBinding, 525
 - wskaźnik myszy, 729
 - wskaźniki graficzne, 292
 - współbieżność, 447
 - współbieżne struktury danych, 499
 - wstawianie danych, 228
 - wstawianie danych do arkusza kalkulacyjnego
 - z pominięciem schowka systemowego, 253
 - wstawianie tekstu do bieżącego dokumentu Worda, 663
 - Wstążka, 687
 - wstrzymywanie działania wątku, 454
 - wtyczki, 595
 - wybór czcionki, 112
 - wybór koloru, 112
 - wybór pliku, 110
 - wycofanie transakcji, 257
 - wygaszacz ekranu, 611
 - wyjątki, 41, 68
 - Exception, 41, 68
 - finally, 69
 - nieobsłużone wyjątki, 69
 - PLINQ, 517
 - przechwytywanie, 42, 68
 - throw, 70
 - try..catch, 41, 42, 68
 - wielokrotne sekcje catch, 68
 - zgłaszanie, 41, 68, 70
 - wykresy, 237, 294, 686
 - konfiguracja, 238
 - seria danych, 238
 - wykrywanie kolizji, 721, 723
 - wyliczenia, 54
 - wyrażenia lambda, 60, 61, 99, 307
 - wyrażenia regularne, 781
 - adres URL, 786
 - cyfry, 782
 - imię żeńskie, 785
 - kod pocztowy, 784
 - modyfikacja łańcucha znakowego, 787
 - odnajdywanie znaków w wyznaczonych miejscach łańcucha, 782
 - odszukiwanie powtarzających się wyrazów, 787
 - określanie ilości wystąpień podanych znaków, 783
 - RegularExpressions, 781
 - tworzenie wzorców, 783
 - znaki, 782
 - znaki specjalne, 781
 - wyróżnianie danych, 242
 - wysyłanie komunikatów Windows, 609
 - wysyłanie komunikatu do okna o znanym uchwycie, 611
 - wyświetlanie
 - obrazy, 713
 - tekst, 747
 - tekst w konsoli, 768
 - wywołania zwrotne, 635
 - wywołanie funkcji Worda, 662
 - wywołanie funkcji z biblioteki DLL, 586
 - wywołanie metody, 50, 55
 - wywołanie metody zdarzeniowej z poziomu kodu, 32
 - wyzwalacze, 187
 - wznawiane działania wątku, 454
- ## X
- XACT, 739
 - AudioEngine, 742
 - banki dźwięków, 739
 - banki wave'ów, 739
 - inicjacja zasobów związanych z dźwiękiem, 743
 - korzystanie z dźwięków, 742
 - LoadContent, 742
 - odtwarzanie dźwięków, 744
 - projekt, 739, 740
 - SoundBank, 742
 - tworzenie bibliotek wave'ów i dźwięków, 740
 - WaveBank, 742
 - XComment, 346
 - XDeclaration, 346
 - XDocument, 346, 347, 348
 - XElement, 346, 347
 - xlApp, 652
 - xlXmlLoadImportToList, 256
 - xlXmlLoadMapXml, 256

xIxmlLoadOpenXml, 256
 xIxmlLoadOption, 255
 xIxmlLoadPromptUser, 256
 XML, 250, 254, 335, 520
 XML Schema, 352
 XML Web Services, 520
 XmlReader, 340
 XmlReaderSettings, 340
 XmlSerializer, 342, 343
 XmlTextReader, 336, 339, 348
 XmlTextWriter, 336, 338, 342
 XNA, 45, 705

- aktualizacja stanu gry, 709, 711
- animacja piłki, 721
- cykl życia gry, 709
- czas, 710
- czcionki, 748
- Direct3D, 755
- dodawanie czcionki do projektu, 748
- dodawanie efektu, 759
- dodawanie graczy, 734
- dodawanie komponentu gry, 731
- Draw, 709, 711, 748
- DrawableGameComponent, 732
- DrawString, 748
- drżenie ekranu, 764
- duszek, 708, 713
- efekty, 755, 758, 762
- Effect, 761
- ElapsedGameTime, 710
- ElapsedTime, 710
- GameComponent, 732
- gamepad, 729
- GamePadState, 729
- GameServices, 733
- GameTime, 710
- generowanie grafiki 2D, 713
- generowanie obrazu, 709
- gra z komputerem, 751
- GraphicsDeviceManager, 708
- HLSL, 756
- Initialize, 710, 732
- IsFixedTimeStep, 710
- IsRunnigSlowly, 710
- klasa gry, 708
- kolizje, 723
- komponenty gry, 731
- LoadContent, 709, 710, 735
- ładowanie czcionki, 749
- ładowanie efektu, 761
- MediaPlayer, 745
- metody gry, 708
- modyfikacja komponentu gry, 732
- obsługa informacji przekazywanej przez użytkownika, 727
- obsługa kolizji, 723, 737, 738
- obsługa kontrolerów gry, 727
- oprawa dźwiękowa, 739
- pętla gry, 709
- PlayerSprite, 733
- poła klasy gry, 708
- potok renderowania, 755

prędkość, 722
 proces renderowania, 755
 projekt gry, 707
 przebiegi, 758
 przetwarzanie pikseli, 755
 rasteryzacja, 755
 rysowanie obrazu, 714
 shadery, 755
 shadery pikseli, 755
 shadery wierzchołków, 755
 Sprite, 714, 717, 722
 SpriteBatch, 708, 711, 714, 733, 748
 stan przycisków myszy, 729
 sterowanie parametrem wejściowym efektu, 761
 sterowanie przy użyciu gamepada, 729
 sterowanie przy użyciu klawiatury, 727
 sterowanie przy użyciu myszy, 728
 TargetElapsedTime, 709
 tekstury, 717
 ticks, 710
 TimeSpan, 710
 TotalGameTime, 710
 transformacja wierzchołków, 755
 tyknięcia, 710
 układ współrzędnych, 719
 UnloadContent, 709
 Update, 709, 711, 721, 732
 wczytywanie obrazu tekstury, 717
 wierzchołki, 755
 wskaźnik myszy, 729
 wykrywanie kolizji, 721, 723, 725, 737
 wyświetlanie obrazu, 713, 718
 wyświetlanie tekstu, 747
 zawartość, 709
 XNA Game Studio 4.0, 705, 707
 XSD, 352

Y

yield, 79, 80

Z

zabezpieczanie kodu pośredniego, 270
 zachłanne ładowanie, 421, 423
 zadania, 469

- Action, 470
- AsParallel, 470
- Cancel, 475
- CancellationToken, 476
- CancellationTokenSource, 476
- ContinueWith, 474, 475
- fabryka zadań, 480
- ForEach, 470
- kontynuowanie zadań, 474
- Parallel, 469
- planista zadań, 482
- przechwytywanie wyjątków, 477
- przekazywanie danych do zadań, 471
- przerywanie oczekiwania na zakończenie zadania, 477
- przerywanie zadań, 475

- przerywanie zadań z uwzględnieniem zmian stanu, 479
- Result, 472
- równoległe wykonanie dużej ilości zadań, 470
- stan zadania, 478
- synchronizacja zadań, 473
- sztafeta zadań, 474
- Task, 469
- TaskContinuationOptions, 475
- TaskCreationOptions, 485
- TaskFactory, 469, 480
- TaskScheduler, 469, 482
- TaskStatus, 478
- test liczby pierwszej, 472
- tworzenie planisty, 483
- tworzenie zadania, 469
- uruchamianie, 470
- ustawienia zadań, 485
- Wait, 470, 473
- WaitAll, 473
- WaitAny, 473
- wstrzymanie bieżącego wątku do momentu zakończenia zadania, 470
- zarządzanie kolejkowaniem zadań, 482
- zwracanie danych z zadania, 472
- Zamknij, 107
- zamrożenie wątku, 499
- zamykanie aplikacji, 107
 - klawisz Esc, 151
- zamykanie serwera automatyzacji MS Excel, 652
- zanikanie obrazu, 762
- zapisywanie
 - dokumenty Worda, 663
 - obrazy do pliku, 131
 - pliki tekstowe, 111
 - pliki XML, 337
 - położenie formy w prywatnym kluczu aplikacji, 573
 - projekt, 24
 - rozmiar formy w prywatnym kluczu aplikacji, 573
 - ustawienia aplikacji, 128
- zapobieganie uruchamianiu wielu instancji aplikacji, 466
- zapytania Entity SQL, 411
- zapytania LINQ, 49, 65, 303, 305, 307, 316, 319, 350, 511
 - źródło danych, 354
- zapytania LINQ to Entities, 410
- zapytania PLINQ, 511
- zapytania T-SQL, 427
- zarządzane biblioteki DLL, 583, 584
 - analiza zawartości biblioteki, 588
 - lista argumentów wybranej metody, 591
 - lista metod w klasie, 589
 - ładowanie dynamiczne, 587
 - odczytywanie zwracanej wartości, 593
 - projekt, 584
 - przekazywanie parametrów, 593
 - tworzenie, 584
 - uruchamianie metody na rzecz instancji obiektu, 593
 - uruchamianie metody statycznej, 593
 - weryfikacja klasy o znanej nazwie, 589
 - weryfikacja obecności metody w klasie, 590
- zarządzanie dodatkami w aplikacji MS Excel, 702
- zarządzanie kolejkowaniem zadań, 482
- zarządzanie transakcjami, 261
- zarządzanie usługą z poziomu innej aplikacji, 646
- zarządzanie wątkiem, 452
- zatrzymanie zadania, 476
- zbiór Julii, 201
- zbiór Mandelbrota, 198, 200
- zdarzenia, 59, 168
 - Click, 169
 - Closed, 150
 - DragDrop, 135, 138
 - DragOver, 135, 138, 139
 - FormClose, 107
 - FormClosed, 128
 - FormClosing, 107
 - interakcja z komponentem, 168
 - KeyPress, 33, 151
 - KeyPreview, 151
 - kontrolki, 29
 - MouseDown, 137
 - MouseMove, 124
 - mysz, 168
 - Paint, 129, 148, 182
 - PrintPage, 114
 - SelectedIndexChanged, 168
 - TextChanged, 108
 - Tick, 126
 - ValueChanged, 29
- zgłaszanie wyjątków, 41, 70
- złośliwe aplikacje, 270
- zmiana
 - ikona okna, 103
 - katalog, 163
 - kolor panelu, 29
 - nazwa okna, 103
 - priorytet wątku, 455
 - stan pozycji menu, 108
 - strumień, 770
 - wartość zmiennej, 47
- zmienne, 47
 - deklaracja, 47
 - dynamiczna zmiana typu, 63
 - inicjacja, 47
 - leniwe inicjowanie, 55
 - określanie typu zmiennej przy inicjacji, 49
 - zmiana wartości, 47
- zmienne środowiskowe, 773
- zmienne wartościowe, 62
- znaki, 48
 - Unicode, 52
- zrównoleglenie pętli, 487
 - CancelationToken, 490
 - for, 487
 - foreach, 489
 - Invoke, 489
 - kontrola wykonywania pętli, 491
 - ParallelLoopResult, 491
 - ParallelLoopState, 491
 - ParallelOptions, 490
 - przerywanie pętli, 490
 - synchronizacja pętli, 492
 - ustawienia pętli, 490
- zrównoległone zapytania LINQ, 511

związki między encjami w zapytaniach, 415
zwracanie tablicy znaków w funkcjach WinAPI, 608
zwracanie wartości przez argumenty, 58, 606

Ż

źródła danych LINQ, 221, 353
drzewa wyrażenia, 354, 361
elementy drzewa wyrażen, 362
IEnumerable, 353
IEnumerable<>, 355

IOrderedQueryable, 359
IOrderedQueryable<>, 360
IQueryable, 354, 359
IQueryable<>, 360
LINQ, 327
LINQ to TXT, 367, 370
oddzielanie źródła od interfejsu, 357
źródło liczb losowych, 353
źródło błędów, 35

Visual Studio 2010 dla programistów C#



Język C# na dobre zadomowił się już na rynku programistycznym. Zarówno on sam, jak i platforma .NET przydają się do najróżniejszych zastosowań, obejmujących nie tylko projektowanie złożonych aplikacji korzystających z baz danych, ale również usług systemu Windows lub gier. Pora więc zapoznać się z tym językiem programowania, oswoić ze środowiskiem Visual Studio 2010 firmy Microsoft i zorientować się, jak możesz wykorzystać nowe umiejętności. Ta obszerna książka o przekrojowym charakterze ułatwi Ci wejście w świat języka C#, platformy .NET i XNA. Poznasz również wiele zaawansowanych technik programistycznych.

W pierwszej części przewodnika poznasz język C# i bogate środowisko programistyczne Visual Studio 2010. Nauczysz się również, jak wydajnie projektować aplikacje z graficznym interfejsem użytkownika. Kolejne części poruszają kwestie połączenia aplikacji z bazą danych i stosowania technologii LINQ. W dobie komputerów z wieloma procesorami o wielu rdzeniach bardzo ważna jest umiejętność tworzenia aplikacji wielowątkowych. Temu zagadnieniu poświęcona jest część czwarta, dotycząca nowej biblioteki TPL. Znajdziesz tu także opis zagadnień związanych z usługami sieciowymi WCF i pracą z wykorzystaniem różnorodnych technologii Windows. Osobną część poświęcono także wprowadzeniu do technologii XNA 4.0 i budowaniu za jej pomocą gier korzystających z grafiki 2D. Jeśli chcesz programować w języku C#, nie możesz przegapić tej książki!

- Środowisko Visual Studio 2010, język C# i debugowanie kodu
- Projektowanie zorientowane obiektowo – przegląd komponentów Windows Forms
- Podstawy ADO.NET, ekspozycja i prezentacja danych, transakcje i raportowanie
- Wprowadzenie do zapytań LINQ – tworzenie próbek danych LINQ
- Programowanie współbieżne – wątki i zadania
- Klasa Parallel i dane w programach równoległych, technologia PLINQ
- Biblioteka WCF – tworzenie usług sieciowych nowej generacji
- Technologie Windows – rejestr systemu, komunikatory, usługi i automatyzacja
- Tworzenie gier w XNA 4.0, użycie chakerów HLSL
- Użycie wyrażen regularnych – sposób na weryfikację wprowadzanych danych
- Tech. jednostkowe – najlepsza metoda sprawdzania poprawności kodu

Nr katalogowy: 5782



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

📖 Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

📰 Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>

Helion SA

ul. Kosciuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

helion.pl
księgarnia
internetowa

Cena 129,00 zł

ISBN 978-83-246-2173-6



9 788324 621736