

John Sharp

Microsoft® Visual C#® 2012

Krok po kroku

Przekład: Janusz Machowski, Krzysztof Szkudlarek

APN Promise, Warszawa 2013

Spis treści

| | |
|---|------|
| <i>Wstęp</i> | xiii |
| Część I: Wprowadzenie do języka Microsoft Visual C# oraz programu Microsoft Visual Studio 2012 | |
| 1 Wprowadzenie do języka C# | 3 |
| Rozpoczynamy programowanie przy użyciu środowiska Visual Studio 2012 | 3 |
| Piszemy pierwszy program | 9 |
| Przestrzenie nazw | 16 |
| Tworzenie aplikacji graficznej | 20 |
| Analiza aplikacji Sklepu Windows | 33 |
| Analiza aplikacji WPF | 38 |
| Dodawanie kodu do aplikacji graficznej | 40 |
| Podsumowanie | 43 |
| Krótki przegląd rozdziału 1 | 44 |
| 2 Zmienne, operatory i wyrażenia | 45 |
| Instrukcje | 45 |
| Identyfikatory | 46 |
| Słowa kluczowe | 47 |
| Zmienne | 48 |
| Nazywanie zmiennych | 48 |
| Deklarowanie zmiennych | 49 |
| Podstawowe typy danych | 50 |
| Zmienne lokalne bez przypisanej wartości | 50 |
| Wyświetlanie wartości podstawowych typów danych | 51 |
| Posługiwanie się operatorami arytmetycznymi | 59 |
| Operatory i typy danych | 60 |
| Poznajemy operatory arytmetyczne | 61 |
| Kontrolowanie pierwszeństwa | 68 |
| Stosowanie zasad łączności przy wyznaczaniu wartości wyrażień | 68 |
| Zasady łączności a operator przypisania | 69 |
| Inkrementacja i dekrementacja wartości zmiennych | 70 |
| Formy przyrostkowe i przedrostkowe | 71 |
| Deklarowanie zmiennych lokalnych o niejawnie określonym typie danych | 72 |
| Podsumowanie | 73 |
| Krótki przegląd rozdziału 2 | 74 |

| | | |
|----------|---|-----|
| 3 | Tworzenie metod i stosowanie zasięgów zmiennych | 75 |
| | Tworzenie metod | 75 |
| | Deklarowanie metody | 76 |
| | Zwracanie danych przez metodę | 77 |
| | Wywoływanie metod | 80 |
| | Stosowanie zasięgu | 82 |
| | Definiowanie zasięgu lokalnego | 83 |
| | Definiowanie zasięgu klasy | 84 |
| | Przeciążanie metod | 84 |
| | Tworzenie metod | 85 |
| | Stosowanie parametrów opcjonalnych oraz nazwanych argumentów | 95 |
| | Definiowanie parametrów opcjonalnych | 97 |
| | Przekazywanie nazwanych argumentów | 97 |
| | Rozwiązywanie niejednoznaczności związanych z parametrami opcjonalnymi i argumentami nazwanymi | 98 |
| | Podsumowanie | 104 |
| | Krótki przegląd rozdziału 3 | 104 |
| 4 | Instrukcje wyboru | 107 |
| | Deklarowanie zmiennych logicznych | 107 |
| | Stosowanie operatorów logicznych | 108 |
| | Operatory równościowe oraz operatory relacji | 108 |
| | Warunkowe operatory logiczne | 109 |
| | Skracanie działania | 110 |
| | Podsumowanie informacji o pierwszeństwie oraz łączności operatorów | 111 |
| | Podjęmowanie decyzji przy użyciu instrukcji <i>if</i> | 112 |
| | Składnia instrukcji <i>if</i> | 112 |
| | Grupowanie instrukcji w bloki | 114 |
| | Kaskadowe łączenie instrukcji <i>if</i> | 115 |
| | Stosowanie instrukcji <i>switch</i> | 121 |
| | Składnia instrukcji <i>switch</i> | 122 |
| | Reguły stosowania instrukcji <i>switch</i> | 123 |
| | Podsumowanie | 127 |
| | Krótki przegląd rozdziału 4 | 128 |
| 5 | Złożone instrukcje przypisania oraz instrukcje iteracji | 129 |
| | Złożone operatory przypisania | 129 |
| | Instrukcja <i>while</i> | 131 |
| | Instrukcja <i>for</i> | 138 |
| | Zasięg instrukcji <i>for</i> | 139 |
| | Instrukcja <i>do</i> | 140 |
| | Podsumowanie | 150 |
| | Krótki przegląd rozdziału 5 | 151 |

| | | |
|----------|---|-----|
| 6 | Obsługa błędów i wyjątków | 153 |
| | Zmaganie się z błędami | 154 |
| | Wypróbowywanie kodu i przechwytywanie wyjątków | 154 |
| | Nieobsłużone wyjątki | 156 |
| | Stosowanie kilku bloków obsługi pułapki | 157 |
| | Przechwytywanie wielu wyjątków | 158 |
| | Propagowanie wyjątków | 165 |
| | Wykonywanie operacji arytmetycznych z kontrolą lub bez kontroli | |
| | przepełnienia | 168 |
| | Pisanie instrukcji objętych kontrolą przepełnienia | 169 |
| | Pisanie wyrażeń objętych kontrolą przepełnienia | 170 |
| | Zgłaszanie wyjątków | 172 |
| | Stosowanie bloku <i>finally</i> | 179 |
| | Podsumowanie | 181 |
| | Krótki przegląd rozdziału 6 | 181 |

Część II: Omówienie modelu obiektowego języka C#

| | | |
|----------|---|-----|
| 7 | Tworzenie i zarządzanie klasami oraz obiektami | 185 |
| | Omówienie klasyfikacji | 186 |
| | Cele hermetyzacji | 186 |
| | Definiowanie i używanie klas | 187 |
| | Kontrolowanie dostępności | 188 |
| | Konstruktory | 190 |
| | Przeciążanie konstruktorów | 192 |
| | Metody i dane statyczne | 201 |
| | Tworzenie pól współdzielonych | 202 |
| | Tworzenie pól statycznych przy użyciu słowa kluczowego <i>const</i> | 203 |
| | Klasy statyczne | 204 |
| | Klasy anonimowe | 207 |
| | Podsumowanie | 208 |
| | Krótki przegląd rozdziału 7 | 209 |
| 8 | Wartości i referencje | 211 |
| | Kopiowanie klas oraz zmiennych typu wartościowego | 211 |
| | Wartości Null oraz typy danych dopuszczające stosowanie wartości Null | 218 |
| | Typy danych dopuszczające stosowanie wartości Null | 220 |
| | Właściwości typów danych dopuszczających stosowanie wartości Null | 221 |
| | Używanie parametrów typu <i>ref</i> i <i>out</i> | 222 |
| | Tworzenie parametrów typu <i>ref</i> | 223 |
| | Tworzenie parametrów typu <i>out</i> | 224 |
| | Sposób organizacji pamięci komputera | 226 |
| | Korzystanie ze stosu oraz ze sterty | 228 |

| | |
|--|------------|
| Klasa <i>System.Object</i> | 229 |
| Opakowywanie typów danych wewnątrz obiektów | 230 |
| Rozpakowywanie typów danych, opakowanych wewnątrz obiektów | 231 |
| Bezpieczne rzutowanie danych | 233 |
| Operator <i>is</i> | 233 |
| Operator <i>as</i> | 233 |
| Podsumowanie | 236 |
| Krótki przegląd rozdziału 8 | 237 |
| 9 Tworzenie typów wartości przy użyciu wyliczeń oraz struktur | 239 |
| Wyliczeniowe typy danych | 239 |
| Deklarowanie wyliczeniowego typu danych | 240 |
| Stosowanie wyliczeniowych typów danych | 240 |
| Wybór wartości literałów wyliczeniowych | 241 |
| Wybór typu danych używanego do wewnętrznego reprezentowania wartości wyliczeniowych | 242 |
| Struktury | 245 |
| Deklarowanie struktury | 247 |
| Omówienie różnic pomiędzy strukturami i klasami | 248 |
| Deklarowanie zmiennych strukturalnych | 250 |
| Omówienie inicjalizacji struktur | 251 |
| Kopiowanie zmiennych strukturalnych | 256 |
| Podsumowanie | 261 |
| Krótki przegląd rozdziału 9 | 261 |
| 10 Tablice | 263 |
| Deklarowanie i tworzenie tablicy | 263 |
| Deklarowanie zmiennych tablicowych | 264 |
| Tworzenie instancji tablicy | 264 |
| Wypełnianie tablic danymi i ich używanie | 266 |
| Tworzenie tablic o niejawnie określonym typie elementów | 267 |
| Korzystanie z indywidualnych elementów tablicy | 268 |
| Wykonywanie iteracji poprzez elementy tablicy | 269 |
| Przekazywanie tablic jako parametrów i zwracanie ich jako wartości metod | 270 |
| Kopiowanie tablic | 273 |
| Tablice wielowymiarowe | 274 |
| Tworzenie tablic nieregularnych | 275 |
| Podsumowanie | 287 |
| Krótki przegląd rozdziału 10 | 288 |
| 11 Tablice parametrów | 289 |
| Przeciążanie: krótkie przypomnienie faktów | 289 |
| Używanie argumentów będących tablicami | 290 |

| | | |
|-----------|--|------------|
| | Deklarowanie tablicy parametrów typu <i>params</i> | 292 |
| | Używanie parametru typu <i>params object[]</i> | 295 |
| | Stosowanie tablicy parametrów typu <i>params</i> | 296 |
| | Porównanie tablic parametrów z parametrami opcjonalnymi | 299 |
| | Podsumowanie | 302 |
| | Krótki przegląd rozdziału 11 | 302 |
| 12 | Dziedziczenie | 303 |
| | Czym jest dziedziczenie? | 303 |
| | Korzystanie z mechanizmów dziedziczenia | 304 |
| | Powtórka informacji na temat klasy <i>System.Object</i> | 306 |
| | Wywoływanie konstruktora klasy bazowej | 307 |
| | Przypisywanie klas | 308 |
| | Deklarowanie metod z użyciem słowa kluczowego <i>new</i> | 310 |
| | Deklarowanie metod wirtualnych | 311 |
| | Deklarowanie metod z użyciem słowa kluczowego <i>override</i> | 313 |
| | Omówienie dostępu chronionego | 316 |
| | Metody rozszerzające | 322 |
| | Podsumowanie | 327 |
| | Krótki przegląd rozdziału 12 | 327 |
| 13 | Tworzenie interfejsów oraz definiowanie klas abstrakcyjnych | 329 |
| | Interfejsy | 330 |
| | Definiowanie interfejsu | 331 |
| | Implementowanie interfejsu | 331 |
| | Odwoływanie się do klasy za pomocą jej interfejsu | 333 |
| | Praca z wieloma interfejsami | 334 |
| | Jawne implementowanie interfejsu | 335 |
| | Ograniczenia interfejsów | 337 |
| | Definiowanie i używanie interfejsów | 338 |
| | Klasy abstrakcyjne | 349 |
| | Metody abstrakcyjne | 350 |
| | Klasy zamknięte | 351 |
| | Metody zamknięte | 351 |
| | Implementowanie i używanie klas abstrakcyjnych | 352 |
| | Podsumowanie | 360 |
| | Krótki przegląd rozdziału 13 | 361 |
| 14 | Proces oczyszczania pamięci i zarządzanie zasobami | 363 |
| | Żywot obiektów | 364 |
| | Tworzenie destruktorów | 365 |
| | Dlaczego istnieje proces oczyszczania pamięci? | 367 |
| | Sposób działania procesu oczyszczania pamięci | 369 |

| | |
|--|-----|
| Zalecenia | 370 |
| Zarządzanie zasobami | 371 |
| Metody sprzątające | 371 |
| Sprzątanie w sposób odporny na występowanie wyjątków | 372 |
| Instrukcja <i>using</i> oraz interfejs <i>IDisposable</i> | 373 |
| Wywoływanie metody <i>Dispose</i> z poziomu destruktora | 374 |
| Implementacja metody sprzątającej w sposób odporny na występowanie wyjątków | 377 |
| Podsumowanie | 386 |
| Krótki przegląd rozdziału 14 | 387 |

Część III: Tworzenie rozszerzalnych typów danych w języku C#

| | |
|--|-----|
| 15 Implementacja właściwości zapewniających dostęp do pól | 391 |
| Implementacja kapsułkowania przy użyciu metod | 391 |
| Co to są właściwości? | 393 |
| Używanie właściwości | 396 |
| Właściwości tylko do odczytu | 396 |
| Właściwości tylko do zapisu | 397 |
| Dostępność właściwości | 397 |
| Ograniczenia właściwości | 398 |
| Deklarowanie właściwości interfejsu | 399 |
| Zastępowanie metod właściwościami | 401 |
| Generowanie automatycznych właściwości | 406 |
| Inicjalizowanie obiektów przy użyciu właściwości | 407 |
| Podsumowanie | 412 |
| Krótki przegląd rozdziału 15 | 412 |
| 16 Indeksatory | 415 |
| Co to jest indeksator? | 415 |
| Przykład bez użycia indeksatorów | 415 |
| Ten sam przykład z wykorzystaniem indeksatorów | 418 |
| Akcesory indeksatora | 420 |
| Indeksatory w interfejsach | 423 |
| Stosowanie indeksatorów w aplikacjach Windows | 424 |
| Podsumowanie | 431 |
| Krótki przegląd rozdziału 16 | 431 |
| 17 Typy ogólne | 433 |
| Problem z typem <i>Object</i> | 433 |
| Rozwiązanie z użyciem typów ogólnych | 437 |
| Typy ogólne a klasy uogólnione | 439 |
| Typy ogólne i nakładanie ograniczeń | 440 |

| | |
|---|------------|
| Tworzenie klasy ogólnej | 440 |
| Teoria drzew binarnych | 440 |
| Budowanie klasy drzewa binarnego przy użyciu typu ogólnego | 444 |
| Tworzenie metody ogólnej | 453 |
| Definiowanie metody ogólnej do budowy drzewa binarnego | 454 |
| Interfejsy ogólne i niezgodność typów | 456 |
| Interfejsy kowariantne | 458 |
| Interfejsy kontrawariantne | 460 |
| Podsumowanie | 462 |
| Krótki przegląd rozdziału 17 | 463 |
| 18 Kolekcje | 465 |
| Co to są klasy kolekcji? | 465 |
| Klasa kolekcji List<T> | 467 |
| Klasa kolekcji LinkedList<T> | 470 |
| Klasa kolekcji Queue<T> | 471 |
| Klasa kolejki Stack<T> | 472 |
| Klasa kolekcji Dictionary<TKey, TValue> | 474 |
| Klasa kolekcji SortedList<TKey, TValue> | 475 |
| Klasa kolekcji HashSet<T> | 476 |
| Inicjalizowanie kolekcji | 478 |
| Metody Find, predykaty i wyrażenia lambda | 479 |
| Porównanie tablic i kolekcji | 481 |
| Wykorzystanie klas kolekcji do gry w karty | 481 |
| Podsumowanie | 486 |
| Krótki przegląd rozdziału 18 | 486 |
| 19 Wyliczanie kolekcji | 489 |
| Wyliczanie elementów kolekcji | 489 |
| Ręczna implementacja modułu wyliczającego | 491 |
| Implementowanie interfejsu IEnumerable | 495 |
| Implementowanie modułu wyliczającego przy użyciu iteratora | 498 |
| Prosty iterator | 498 |
| Definiowanie modułu wyliczającego dla klasy Tree<TItem> przy użyciu iteratora | 500 |
| Podsumowanie | 503 |
| Krótki przegląd rozdziału 19 | 503 |
| 20 Wydzielanie logiki aplikacji i obsługa zdarzeń | 505 |
| Co to są delegaty | 506 |
| Przykłady delegatów w bibliotece klas .NET Framework | 507 |
| Przykład zautomatyzowanej fabryki | 508 |
| Implementowanie systemu sterowania fabryką bez użycia delegatów | 510 |

| | |
|---|------------|
| Implementowanie sterowania fabryką przy użyciu delegata | 510 |
| Deklarowanie i używanie delegatów. | 513 |
| Delegaty i wyrażenia lambda. | 522 |
| Tworzenie adaptera metody. | 523 |
| Różne formy wyrażen lambda | 524 |
| Włączanie powiadomień za pomocą zdarzeń | 526 |
| Deklarowanie zdarzenia | 526 |
| Subskrypcja zdarzenia | 527 |
| Anulowanie subskrypcji zdarzenia | 528 |
| Zgłaszanie zdarzenia | 528 |
| Zdarzenia interfejsu użytkownika | 529 |
| Używanie zdarzeń | 530 |
| Podsumowanie | 537 |
| Krótki przegląd rozdziału 20. | 538 |
| 21 Odpytywanie danych w pamięci przy użyciu wyrażen w języku zapytań | 541 |
| Co to jest LINQ (Language-Integrated Query)? | 542 |
| Używanie LINQ w aplikacjach C# | 543 |
| Wybieranie danych | 544 |
| Filtrowanie danych. | 547 |
| Porządkowanie, grupowanie i agregowanie danych | 548 |
| Łączenie danych. | 550 |
| Operatory zapytań. | 551 |
| Odpytywanie danych w obiektach Tree<TItem> | 554 |
| LINQ i opóźnione przetwarzanie. | 560 |
| Podsumowanie | 564 |
| Krótki przegląd rozdziału 21 | 564 |
| 22 Przeciążanie operatorów | 567 |
| Czym są operatory. | 567 |
| Ograniczenia operatorów | 568 |
| Operatory przeciążone | 569 |
| Tworzenie operatorów symetrycznych | 570 |
| Jak wygląda przetwarzanie złożonej instrukcji przypisania | 572 |
| Deklarowanie operatorów zwiększających i zmniejszających. | 573 |
| Operatory porównań w strukturach i klasach. | 574 |
| Definiowanie par operatorów | 575 |
| Implementowanie operatorów | 576 |
| Operatory konwersji | 583 |
| Wbudowane metody konwersji. | 583 |
| Implementowanie własnych operatorów konwersji | 584 |
| Tworzenie operatorów symetrycznych – uzupełnienie | 585 |

| | |
|--|-----|
| Zapisywanie operatorów konwersji | 586 |
| Podsumowanie | 589 |
| Krótki przegląd rozdziału 22 | 589 |

Część IV: Tworzenie profesjonalnych aplikacji w C# dla Windows 8

| | | |
|-----------|---|------------|
| 23 | Przyspieszanie działania za pomocą zadań | 593 |
| | Po co stosować wielozadaniowość przy użyciu przetwarzania równoległego? .. | 593 |
| | Narodziny procesora wielordzeniowego | 595 |
| | Implementowanie wielozadaniowości w .NET Framework | 596 |
| | Zadania, wątki i pula wątków. | 597 |
| | Tworzenie, uruchamianie i kontrolowanie zadań | 598 |
| | Używanie klasy Task do implementacji równoległości | 601 |
| | Tworzenie abstrakcji zadań za pomocą klasy Parallel | 612 |
| | Kiedy nie używać klasy Parallel | 617 |
| | Anulowanie zadań i obsługa wyjątków | 619 |
| | Mechanizm anulowania kooperatywnego | 619 |
| | Kontynuowanie w przypadku zadań anulowanych lub przerwanych z powodu wyjątku | 631 |
| | Podsumowanie | 634 |
| | Krótki przegląd rozdziału 23 | 634 |
| 24 | Skracanie czasu reakcji za pomocą działań asynchronicznych | 637 |
| | Implementowanie metod asynchronicznych | 638 |
| | Definiowanie metod asynchronicznych: postawienie problemu | 639 |
| | Definiowanie metod asynchronicznych: rozwiązanie | 642 |
| | Definiowanie metod asynchronicznych zwracających wartości | 648 |
| | Metody asynchroniczne i interfejsy API środowiska Windows Runtime | 649 |
| | Zrównoleglanie deklaratywnego dostępu do danych za pomocą PLINQ | 653 |
| | Wykorzystanie PLINQ do poprawy wydajności podczas wykonywania iteracji po elementach kolekcji | 653 |
| | Anulowanie zapytania PLINQ | 658 |
| | Synchronizowanie współbieżnych operacji dostępu do danych | 659 |
| | Blokowanie danych | 662 |
| | Elementarne narzędzia synchronizacji umożliwiające koordynowanie zadań | 662 |
| | Anulowanie synchronizacji | 665 |
| | Współbieżne klasy kolekcji | 666 |
| | Wykorzystanie kolekcji współbieżnej i blokady do implementacji dostępu do danych przystosowanego do trybu wielowątkowego | 667 |
| | Podsumowanie | 677 |
| | Krótki przegląd rozdziału 24 | 678 |

| | | |
|-----------|--|------------|
| 25 | Implementowanie interfejsu użytkownika aplikacji Sklepu Windows | 681 |
| | Co to jest aplikacja Sklepu Windows? | 682 |
| | Budowa aplikacji Sklepu Windows przy użyciu szablonu Blank App | 686 |
| | Implementowanie skalowalnego interfejsu użytkownika | 689 |
| | Stosowanie stylów do interfejsu użytkownika | 720 |
| | Podsumowanie | 730 |
| | Krótki przegląd rozdziału 25 | 731 |
| 26 | Wyświetlanie i wyszukiwanie danych w aplikacjach Sklepu Windows | 733 |
| | Implementowanie wzorca projektowego Model-View-ViewModel | 734 |
| | Wyświetlanie danych przy użyciu mechanizmu wiązania danych | 735 |
| | Modyfikowanie danych przy użyciu mechanizmu wiązania danych | 741 |
| | Stosowanie wiązania danych do kontrolki ComboBox | 745 |
| | Tworzenie składnika ViewModel | 748 |
| | Dodawanie poleceń do składnika ViewModel | 752 |
| | Kontrakty systemu Windows 8 | 763 |
| | Implementowanie kontraktu Search | 765 |
| | Nawigowanie do wybranego elementu | 775 |
| | Uruchamianie aplikacji za pomocą elementu Search paska Charms | 778 |
| | Podsumowanie | 782 |
| | Krótki przegląd rozdziału 26 | 785 |
| 27 | Dostęp do zdalnej bazy danych z poziomu aplikacji Sklepu Windows | 787 |
| | Pobieranie danych z bazy danych | 788 |
| | Utworzenie modelu encji | 789 |
| | Tworzenie i wykorzystywanie usługi danych | 796 |
| | Wstawianie, aktualizowanie i usuwanie informacji w bazie danych | 809 |
| | Wykonywanie operacji wstawiania, aktualizowania i usuwania danych za pośrednictwem usługi danych WCF | 809 |
| | Raportowanie błędów i aktualizowanie interfejsu użytkownika | 820 |
| | Podsumowanie | 828 |
| | Krótki przegląd rozdziału 27 | 829 |
| | Indeks | 831 |