

Przedmowa (17)

1. Liczby i typy wyliczeniowe (27)

- 1.0. Wprowadzenie (27)
- 1.1. Określanie przybliżonej równości pomiędzy wartością ułamkową a zmiennoprzecinkową (28)
- 1.2. Konwersja stopni na radiany (30)
- 1.3. Konwersja radianów na stopnie (31)
- 1.4. Zastosowanie operatora negacji bitowej do danych różnych typów (31)
- 1.5. Sprawdzenie, czy liczba jest parzysta czy nieparzysta (33)
- 1.6. Uzyskanie bardziej znaczącego i mniej znaczącego słowa liczby (34)
- 1.7. Konwersja liczby z innego systemu liczbowego na dziesiętny (35)
- 1.8. Sprawdzenie, czy ciąg znaków reprezentuje prawidłową liczbę (37)
- 1.9. Zaokrąglanie wartości zmiennoprzecinkowych (37)
- 1.10. Wybór algorytmu zaokrąglania (38)
- 1.11. Zamiana stopni Celsjusza na stopnie Fahrenheita (39)
- 1.12. Zamiana stopni Fahrenheita na stopnie Celsjusza (40)
- 1.13. Bezpieczna konwersja liczb do mniejszego rozmiaru (40)
- 1.14. Wyznaczanie długości dowolnego z trzech boków trójkąta prostokątnego (43)
- 1.15. Obliczanie kątów trójkąta prostokątnego (44)
- 1.16. Wyświetlanie wartości typu wyliczeniowego w postaci tekstowej (45)
- 1.17. Konwersja zwykłego tekstu na odpowiedniki w postaci wartości typu wyliczeniowego (47)
- 1.18. Sprawdzanie poprawności wartości typu wyliczeniowego (48)
- 1.19. Sprawdzanie poprawności typu wyliczeniowego z atrybutem Flags (50)
- 1.20. Zastosowanie elementów typu wyliczeniowego w masce bitowej (52)
- 1.21. Sprawdzanie, czy ustawiono jedną czy kilka flag w danych typu wyliczeniowego (55)
- 1.22. Wyznaczanie części całkowitej zmiennej typu decimal lub double (58)

2. Znaki i ciągi znaków (59)

- 2.0. Wprowadzenie (59)
- 2.1. Określenie rodzaju znaku w zmiennej char (59)
- 2.2. Sprawdzanie, czy znak znajduje się w określonym zakresie (62)
- 2.3. Porównywanie znaków z rozróżnianiem wielkości liter i bez niego (63)
- 2.4. Wyszukiwanie wszystkich wystąpień znaku w ciągu (65)
- 2.5. Wyszukiwanie wszystkich wystąpień jednego ciągu znaków w innym (67)
- 2.6. Implementacja prostego analizatora dzielącego tekst na słowa (71)
- 2.7. Zarządzanie rozróżnianiem wielkich i małych liter podczas porównywania dwóch ciągów znaków (73)
- 2.8. Porównywanie jednego ciągu znaków z początkiem lub końcem innego (74)
- 2.9. Wstawianie tekstu do ciągu znaków (75)
- 2.10. Usuwanie lub zastępowanie znaków w ciągu (76)
- 2.11. Kodowanie danych binarnych w formacie Base64 (78)
- 2.12. Dekodowanie danych binarnych zakodowanych w formacie Base64 (79)
- 2.13. Konwersja obiektu String zwróconego w formacie Byte[] na postać String (80)
- 2.14. Przekazywanie ciągu znaków do metody, która akceptuje wyłącznie dane typu byte[] (82)
- 2.15. Konwersja ciągów znaków na dane innych typów (83)

- 2.16. Formatowanie danych w ciągach znaków (86)
- 2.17. Tworzenie ciągu znaków rozdzielanego separatorami (89)
- 2.18. Wyodrębnianie elementów z tekstu rozdzielanego separatorami (90)
- 2.19. Ustawienie maksymalnej liczby znaków dla obiektów klasy StringBuilder (91)
- 2.20. Przetwarzanie w pętli wszystkich znaków w ciągu (92)
- 2.21. Poprawa wydajności porównywania ciągów znaków (94)
- 2.22. Poprawa wydajności aplikacji wykorzystujących klasę StringBuilder (97)
- 2.23. Usuwanie znaków z początku i (lub) końca ciągu (100)
- 2.24. Sprawdzanie, czy ciąg znaków jest pusty lub zawiera wartość null (101)
- 2.25. Dołączanie wiersza (101)
- 2.26. Kodowanie danych przekazywanych porcjami (102)

3. Klasy i struktury (109)

- 3.0. Wprowadzenie (109)
- 3.1. Tworzenie struktur działających jak unie (111)
- 3.2. Wyprowadzanie wartości typu w postaci ciągu znaków (113)
- 3.3. Konwersja znakowej reprezentacji obiektu na obiekt (118)
- 3.4. Implementacja polimorfizmu za pomocą abstrakcyjnych klas bazowych (122)
- 3.5. Zapewnienie możliwości sortowania danych zdefiniowanego typu (127)
- 3.6. Zapewnienie możliwości wyszukiwania danych typu (132)
- 3.7. Pośrednie przeciążanie operatorów +=, -=, /= i *= (136)
- 3.8. Pośrednie przeciążanie operatorów &&, || i ?: (139)
- 3.9. Włączanie i wyłączanie bitów (141)
- 3.10. Tworzenie bezbłędnych wyrażeń (145)
- 3.11. Upraszczanie wyrażeń logicznych (147)
- 3.12. Konwersja prostych typów danych w sposób niezależny od języka (150)
- 3.13. Kiedy należy używać operatora cast, a kiedy as lub is? (156)
- 3.14. Konwersja za pomocą operatora as (157)
- 3.15. Sprawdzanie typu zmiennej za pomocą operatora is (159)
- 3.16. Implementacja polimorfizmu za pomocą interfejsów (162)
- 3.17. Wywoływanie tej samej metody dla wielu typów obiektowych (165)
- 3.18. Implementacja wywoływanej zwrótnie metody powiadamiającej z wykorzystaniem interfejsów (167)
- 3.19. Wykorzystanie wielu punktów wejścia w celu stworzenia kilku wersji aplikacji (175)
- 3.20. Zapobieganie tworzeniu częściowo zainicjowanych obiektów (176)
- 3.21. Zwracanie wielu elementów przez metodę (178)
- 3.22. Analiza parametrów wiersza polecenia (181)
- 3.23. Przystosowanie klasy do współpracy z obiektami COM (188)
- 3.24. Inicjowanie stałej w fazie wykonywania programu (192)
- 3.25. Pisanie kodu zgodnego z jak największą liczbą zarządzanych języków (195)
- 3.26. Tworzenie klas, które można klonować (196)
- 3.27. Zapewnienie niszczenia obiektu (199)
- 3.28. Zwalnianie obiektu COM z poziomu zarządzanego kodu (202)
- 3.29. Tworzenie pamięci podręcznej obiektów (203)
- 3.30. Wycofywanie zmian wprowadzonych w obiektach (212)
- 3.31. Zwalnianie niezarządzanych zasobów (218)
- 3.32. Wyszukiwanie operacji pakowania i rozpakowania (224)

4. Typy generyczne (227)

- 4.0. Wprowadzenie (227)
- 4.1. Gdzie i kiedy korzystać z typów generycznych? (227)
- 4.2. Podstawowe wiadomości o typach generycznych (228)
- 4.3. Odczytywanie obiektu Type dla danych typu generycznego (235)
- 4.4. Zastępowanie typu ArrayList jego generycznym odpowiednikiem (236)
- 4.5. Zastąpienie obiektów Stack i Queue ich generycznymi odpowiednikami (240)
- 4.6. Implementacja powiązanych list (244)
- 4.7. Tworzenie typu wartości, który można zainicjować wartością null (247)
- 4.8. Odwrócenie porządku posortowanej listy (249)
- 4.9. Tworzenie kolekcji tylko do odczytu z wykorzystaniem typów generycznych (271)
- 4.10. Zastąpienie typu Hashtable jego generycznym odpowiednikiem (273)
- 4.11. Korzystanie z pętli foreach dla generycznego typu Dictionary (276)
- 4.12. Ograniczenia dla argumentów opisujących typy (277)
- 4.13. Inicjowanie zmiennych generycznych na ich wartości domyślne (279)

5. Kolekcje (281)

- 5.0. Wprowadzenie (281)
- 5.1. Zamiana miejscami dwóch elementów w tablicy (283)
- 5.2. Szybkie odwracanie tablicy (284)
- 5.3. Odwracanie tablic dwuwymiarowych (286)
- 5.4. Odwracanie tablic postrzępionych (288)
- 5.5. Bardziej uniwersalna klasa StackTrace (289)
- 5.6. Określanie liczby wystąpień elementu na liście List<T> (294)
- 5.7. Wyodrębnianie wszystkich egzemplarzy określonego elementu z listy List<T> (297)
- 5.8. Wstawianie i usuwanie elementów z tablicy (300)
- 5.9. Utrzymywanie listy List<T> w stanie posortowanym (302)
- 5.10. Sortowanie indeksów i (lub) wartości obiektu klasy Dictionary (304)
- 5.11. Tworzenie obiektu Dictionary z ograniczeniami dla wartości minimalnej i maksymalnej (307)
- 5.12. Wyświetlanie danych z tablicy w postaci ciągu znaków rozdzielanych separatorami (310)
- 5.13. Zapisywanie migawek list w tablicy (311)
- 5.14. Utrzymywanie kolekcji pomiędzy sesjami aplikacji (312)
- 5.15. Sprawdzanie wszystkich elementów tablicy Array bądź List<T> (314)
- 5.16. Wykonywanie operacji dla każdego elementu danych typu Array bądź List<T> (315)
- 5.17. Tworzenie obiektów tylko do odczytu typu Array lub List<T> (317)

6. Iteratory i typy częściowe (319)

- 6.0. Wprowadzenie (319)
- 6.1. Implementacja zagnieżdżonych pętli foreach dla klasy (320)
- 6.2. Tworzenie własnej obsługi pętli foreach (324)
- 6.3. Tworzenie iteratorów dla typu generycznego (327)
- 6.4. Tworzenie iteratora dla typu niegenerycznego (329)

- 6.5. Tworzenie iteratorów z parametrami (331)
- 6.6. Definiowanie wielu iteratorów dla jednego typu (333)
- 6.7. Implementacja iteratorów jako operatorów przeciążonych (336)
- 6.8. Wymuszone zatrzymywanie iteratora (342)
- 6.9. Obsługa bloku finally w iteratorach (344)
- 6.10. Organizacja implementacji interfejsów (347)
- 6.11. Generowanie kodu spoza głównej ścieżki (351)

7. Obsługa wyjątków (355)

- 7.0. Wprowadzenie (355)
- 7.1. Weryfikacja parametrów kluczowych (361)
- 7.2. Gdzie należy przechwytywać i ponawiać zgłaszanie wyjątków? (364)
- 7.3. Identyfikacja wyjątków i ich zastosowanie (365)
- 7.4. Indywidualna obsługa wyjątków pochodnych (365)
- 7.5. Jak zapewnić, aby wyjątki nie były tracone w przypadku wykorzystania bloków finally? (369)
- 7.6. Obsługa wyjątków zgłaszanych przez metody wywoływane za pomocą odbić (372)
- 7.7. Diagnostowanie problemów podczas ładowania kompilatów (374)
- 7.8. Odzworowania pomiędzy zarządzanymi wyjątkami a wartościami HRESULT (376)
- 7.9. Obsługa wartości HRESULT definiowanych przez użytkownika (379)
- 7.10. Przeciwdziałanie nieobsłużonym wyjątkom (380)
- 7.11. Uzyskiwanie informacji o wyjątkach (382)
- 7.12. Jak szybko dostać się do meritum problemu? (385)
- 7.13. Tworzenie własnych typów opisu wyjątków (386)
- 7.14. Odczytywanie obrazu stosu (397)
- 7.15. Ustawienie pułapki w miejscu, gdzie może wystąpić wyjątek "pierwszej szansy" (399)
- 7.16. Zapobieganie wyjątkowi TypeInitializationException (401)
- 7.17. Obsługa wyjątków zgłaszanych przez delegaty asynchroniczne (405)
- 7.18. Przekazywanie do wyjątków dodatkowych informacji za pośrednictwem pola Exception.Data (406)
- 7.19. Prezentacja wyjątków w sposób niestandardowy za pomocą wizualizatorów (408)
- 7.20. Postępowanie z nieobsłużonymi wyjątkami w aplikacjach WinForms (414)

8. Diagnostyka (417)

- 8.0. Wprowadzenie (417)
- 8.1. Zarządzanie wynikami diagnostycznymi we wdrożonych aplikacjach (418)
- 8.2. Szczegółowe zarządzanie wynikami debugowania (śledzenia) (421)
- 8.3. Tworzenie własnych klas przełączników (424)
- 8.4. Warunkowa kompilacja bloków kodu (428)
- 8.5. Jak sprawdzić, czy proces przestał odpowiadać? (430)
- 8.6. Wykorzystanie dzienników zdarzeń w aplikacji (432)
- 8.7. Modyfikacja maksymalnego rozmiaru niestandardowego dziennika zdarzeń (438)
- 8.8. Wyszukiwanie zapisów w dzienniku zdarzeń (439)
- 8.9. Obserwacja specyficznego zapisu w dzienniku zdarzeń (443)

- 8.10. Wyszukiwanie wszystkich źródeł należących do określonego dziennika zdarzeń (444)
- 8.11. Implementacja prostego licznika wydajności (446)
- 8.12. Implementacja liczników wydajności, które wymagają liczników bazowych (449)
- 8.13. Włączanie i wyłączanie złożonego kodu śledzenia (452)
- 8.14. Przechwytywanie standardowego wyniku procesu (455)
- 8.15. Tworzenie niestandardowego wyjścia debugowania dla klas użytkownika (457)
- 8.16. Odczytywanie ustawień bieżącej domeny AppDomain (459)
- 8.17. Programowe podwyższanie priorytetu procesu (462)
- 8.18. Analiza środowiska wykonawczego w celu diagnozowania problemów (463)

9. Delegaty, zdarzenia i metody anonimowe (465)

- 9.0. Wprowadzenie (465)
- 9.1. Zarządzanie czasem i miejscem uruchomienia delegatu w obrębie delegatu multicast (466)
- 9.2. Odczytywanie zwracanych wyników każdego z delegatów wchodzących w skład delegatu multicast (469)
- 9.3. Indywidualna obsługa wyjątków dla każdego z delegatów w obrębie delegatu multicast (471)
- 9.4. Konwersja wywołania delegatu z synchronicznego na asynchroniczne (474)
- 9.5. Opakowywanie zapieczętowanych klas w celu dodawania zdarzeń (477)
- 9.6. Przekazywanie specjalizowanych parametrów do zdarzenia i ze zdarzenia (483)
- 9.7. Zaawansowany mechanizm wyszukiwania interfejsów (488)
- 9.8. Zaawansowany mechanizm wyszukiwania składowych (491)
- 9.9. Obserwacja dodawania i modyfikowania elementów w tablicy Hashtable (495)
- 9.10. Wykorzystanie "haków" do klawiszy Windows (502)
- 9.11. Śledzenie operacji wykonywanych myszą i reagowanie na nie (508)
- 9.12. Zastosowanie metod anonimowych (509)
- 9.13. Lepsza konfiguracja metod obsługi zdarzeń (513)
- 9.14. Wykorzystywanie różnych modyfikatorów parametrów w metodach anonimowych (516)
- 9.15. Zastosowanie domknięć w języku C# (519)
- 9.16. Wykonywanie wielu operacji na liście z wykorzystaniem funktorów (523)

10. Wyrażenia regularne (527)

- 10.0. Wprowadzenie (527)
- 10.1. Przetwarzanie ciągów spełniających warunki wyrażenia regularnego (528)
- 10.2. Wyodrębnianie grup z obiektu MatchCollection (531)
- 10.3. Weryfikacja składni wyrażenia regularnego (533)
- 10.4. Szybki sposób wyszukiwania ostatniego podciągu spełniającego kryteria (535)
- 10.5. Zastępowanie znaków lub słów w ciągu znaków (536)
- 10.6. Ulepszanie prostej funkcji do zastępowania ciągów znaków (539)
- 10.7. Implementacja lepszego tokenizera (542)
- 10.8. Kompilacja wyrażeń regularnych (543)
- 10.9. Zliczanie wierszy tekstu (545)
- 10.10. Zwracanie całych wierszy w przypadku znalezienia podciągu pasującego do wzorca (548)

- 10.11. Wyszukiwanie określonego wystąpienia pasującego podciągu (551)
- 10.12. Wykorzystanie często używanych wzorców (553)
- 10.13. Dokumentowanie wyrażeń regularnych (556)
- 10.14. Zastosowanie wbudowanych wyrażeń regularnych do analizy stron ASP.NET (557)

11. Algorytmy i struktury danych (563)

- 11.0. Wprowadzenie (563)
- 11.1. Tworzenie skrótów dla typów danych (563)
- 11.2. Tworzenie kolejek z priorytetami (571)
- 11.3. Tworzenie kolejek dwukierunkowych (578)
- 11.4. Sprawdzanie zrównoważenia znaków lub ciągów (584)
- 11.5. Tworzenie odwzorowania jeden do wielu (588)
- 11.6. Tworzenie drzew binarnych (596)
- 11.7. Tworzenie drzewa n-arnego (608)
- 11.8. Tworzenie obiektu Set (619)

12. Operacje wejścia-wyjścia w systemie plików (631)

- 12.0. Wprowadzenie (631)
- 12.1. Tworzenie, kopiowanie, przenoszenie lub usuwanie pliku (632)
- 12.2. Operacje na atrybutach plików (634)
- 12.3. Zmiana nazwy pliku (637)
- 12.4. Ustalanie, czy plik istnieje (638)
- 12.5. Wybór metody otwarcia pliku lub strumienia dla zapisu i (lub) odczytu (639)
- 12.6. Losowy dostęp do części pliku (645)
- 12.7. Generowanie znaku EOL niezależnego od platformy (649)
- 12.8. Tworzenie pliku, zapisywanie do niego i odczytywanie z niego (650)
- 12.9. Ustalanie, czy istnieje katalog (657)
- 12.10. Tworzenie, kopiowanie, przenoszenie i usuwanie katalogu (657)
- 12.11. Operacje na atrybutach katalogów (659)
- 12.12. Zmiana nazwy katalogu (662)
- 12.13. Wyszukiwanie katalogów lub plików przy użyciu symboli wieloznacznych (663)
- 12.14. Odczytywanie drzewa katalogów (667)
- 12.15. Parsowanie ścieżki dostępu (669)
- 12.16. Parsowanie ścieżek dostępu w zmiennych środowiskowych (671)
- 12.17. Weryfikacja ścieżki dostępu (672)
- 12.18. Używanie w aplikacji pliku tymczasowego (676)
- 12.19. Otwieranie strumienia pliku przy użyciu jedynie uchwytu pliku (677)
- 12.20. Jednoczesne zapisywanie do wielu plików wyjściowych (679)
- 12.21. Uruchamianie i używanie narzędzi konsoli (682)
- 12.22. Blokowanie części pliku (683)
- 12.23. Wyszukiwanie w systemie plików konkretnych zmian w jednym lub więcej plikach lub katalogach (686)
- 12.24. Oczekiwanie na wykonanie określonej czynności w systemie plików (691)
- 12.25. Porównywanie wersji dwóch modułów wykonywalnych (694)
- 12.26. Uzyskiwanie informacji o wszystkich napędach obecnych w systemie (697)
- 12.27. Szyfrowanie i deszyfrowanie istniejącego pliku (700)

- 12.28. Kompresowanie i dekompresja plików (701)

13. Odzwierciedlanie (705)

- 13.0. Wprowadzenie (705)
- 13.1. Odczytywanie listy podzespółów zależnych (705)
- 13.2. Odczytywanie listy eksportowanych typów (708)
- 13.3. Odnajdywanie metod pokrytych (709)
- 13.4. Odnajdywanie składowych w podzespole (713)
- 13.5. Odnajdywanie składowych w interfejsie (715)
- 13.6. Ustalanie i odczytywanie typów zagnieżdżonych znajdujących się w podzespole (716)
- 13.7. Wyświetlanie hierarchii dziedziczenia typu (718)
- 13.8. Odnajdywanie podklas typu (720)
- 13.9. Odnajdywanie w podzespole wszystkich typów, które można serializować (721)
- 13.10. Filtrowanie danych w trakcie odczytywania składowych (723)
- 13.11. Dynamiczne wywoływanie składowych (727)
- 13.12. Definiowanie wskazówek dla zaciemniaczy kodu (730)
- 13.13. Ustalanie, czy typ lub metoda ma charakter ogólny (732)
- 13.14. Odczytywanie manifestu zasobów w kodzie źródłowym (734)
- 13.15. Odczytywanie informacji o zmiennych lokalnych (735)
- 13.16. Tworzenie typu ogólnego (737)

14. Sieć WWW (739)

- 14.0. Wprowadzenie (739)
- 14.1. Odczytywanie nazwy komputera na podstawie adresu IP (739)
- 14.2. Odczytywanie adresu IP komputera o podanej nazwie (740)
- 14.3. Parsowanie URI (741)
- 14.4. Formowanie i weryfikacja URI bezwzględnego (744)
- 14.5. Obsługa błędów serwera WWW (746)
- 14.6. Komunikacja z serwerem WWW (748)
- 14.7. Przesyłanie żądań przez serwer proxy (750)
- 14.8. Odczytywanie kodu HTML z podanego adresu URL (751)
- 14.9. Wykorzystanie nowej kontrolki przeglądarki internetowej (753)
- 14.10. Wiązanie tabel baz danych z pamięcią podręczną (755)
- 14.11. Zapisywanie w pamięci podręcznej danych z wieloma powiązaniem (756)
- 14.12. Prekompilacja strony ASP.NET z poziomu kodu źródłowego (758)
- 14.13. Uwzględnianie i pomijanie sekwencji ucieczki w danych dla sieci WWW (761)
- 14.14. Wykorzystanie klasy UriBuilder (763)
- 14.15. Analiza i zmiana konfiguracji aplikacji sieciowej (765)
- 14.16. Praca z kodem HTML (767)
- 14.17. Zwiększanie wydajności pracy z HTTP przez zapisywanie wyników w pamięci podręcznej (770)
- 14.18. Sprawdzanie własnych stron obsługi błędów używanych przez serwer (771)
- 14.19. Odczytywanie odwzorowań aplikacji dla ASP.NET zdefiniowanych na serwerze IIS (774)

15. XML (777)

- 15.0. Wprowadzenie (777)
- 15.1. Wczytywanie i dostęp do danych XML w kolejności wyznaczonej w dokumencie (777)
- 15.2. Odczyt dokumentu XML z sieci WWW (780)
- 15.3. Wyszukiwanie informacji w dokumencie XML (782)
- 15.4. Weryfikacja poprawności danych XML (784)
- 15.5. Tworzenie dokumentu XML z poziomu kodu źródłowego (789)
- 15.6. Wykrywanie zmian w dokumencie XML (791)
- 15.7. Obsługa niedozwolonych znaków w ciągu znaków XML (794)
- 15.8. Przekształcanie danych XML (796)
- 15.9. Dzielenie dokumentu XML na części (800)
- 15.10. Składanie dokumentu XML z części (804)
- 15.11. Weryfikacja poprawności zmienionego dokumentu XML bez jego ponownego ładowania (808)
- 15.12. Rozszerzanie przekształceń XSLT (810)
- 15.13. Odczytywanie schematu z istniejących plików XML (813)
- 15.14. Przekazywanie parametrów do transformacji XSLT (815)

16. Praca w sieci (819)

- 16.0. Wprowadzenie (819)
- 16.1. Tworzenie serwera TCP (819)
- 16.2. Tworzenie klienta TCP (824)
- 16.3. Symulowanie przetwarzania formularza (827)
- 16.4. Pobieranie danych z serwera (830)
- 16.5. Komunikacja przy użyciu potoków nazwanych (831)
- 16.6. Pingowanie z poziomu kodu źródłowego (850)
- 16.7. Wysyłanie poczty SMTP przy użyciu usługi SMTP (852)
- 16.8. Sprawdzanie parametrów dostępu do sieci (856)
- 16.9. Skanowanie portów komputera przy użyciu gniazd (861)
- 16.10. Używanie bieżących ustawień połączenia z Internetem (865)
- 16.11. Pobieranie pliku za pośrednictwem FTP (871)

17. Bezpieczeństwo (873)

- 17.0. Wprowadzenie (873)
- 17.1. Kontrola dostępu do typów w podzespole lokalnym (873)
- 17.2. Szyfrowanie i rozszyfrowywanie ciągu znaków (881)
- 17.3. Szyfrowanie i rozszyfrowywanie pliku (885)
- 17.4. Usuwanie danych dotyczących szyfrowania (889)
- 17.5. Sprawdzenie, czy ciąg znaków nie uległ uszkodzeniu w trakcie transmisji (892)
- 17.6. Przesłanie mechanizmu dodającego wartość mieszającą do ciągu znaków (895)
- 17.7. Ulepszony generator liczb losowych (900)
- 17.8. Bezpieczne przechowywanie danych (901)
- 17.9. Zabezpieczanie asertacji bezpieczeństwa (907)
- 17.10. Zapobieganie niepożądanym zmianom w podzespole (909)
- 17.11. Sprawdzanie, czy podzespołowi nadano odpowiednie uprawnienia (912)
- 17.12. Minimalizowanie zakresu uprawnień podzespołu umożliwiających przeprowadzenie ataku (913)

- 17.13. Uzyskiwanie informacji dotyczących monitorowania i zabezpieczeń (914)
- 17.14. Nadawanie i odbieranie dostępu do pliku lub klucza rejestru (919)
- 17.15. Zabezpieczanie danych w postaci ciągów znaków (921)
- 17.16. Zabezpieczanie strumienia danych (924)
- 17.17. Szyfrowanie danych w pliku web.config (931)
- 17.18. Rozpoznawanie pełnej przyczyny zgłoszenia wyjątku SecurityException (933)
- 17.19. Zabezpieczanie procesu kodowania Unicode (935)
- 17.20. Pozyskiwanie bezpieczniejszego uchwytu pliku (936)

18. Wątki i synchronizacja (939)

- 18.0. Wprowadzenie (939)
- 18.1. Tworzenie pól statycznych dla konkretnych wątków (939)
- 18.2. Zapewnianie dostępu o bezpiecznych wątkach do składowych klasy (942)
- 18.3. Zapobieganie cichemu zakończeniu wątków (947)
- 18.4. Odpytywanie asynchronicznej metody delegowanej (949)
- 18.5. Definiowanie czasu wygasania asynchronicznej metody delegowanej (952)
- 18.6. Uzyskiwanie powiadomienia o zakończeniu działania asynchronicznej metody delegowanej (954)
- 18.7. Ustalanie, czy żądanie skierowane do puli wątków zostanie zakolejkowane (957)
- 18.8. Konfigurowanie licznika czasu (959)
- 18.9. Bezpieczne przechowywanie danych wątku (962)
- 18.10. Przydzielanie dostępu do zasobu więcej niż jednemu klientowi przy użyciu semafora (965)
- 18.11. Synchronizowanie wielu procesów przy użyciu muteksu (969)
- 18.12. Zapewnianie współpracy między wątkami za pomocą zdarzeń (979)
- 18.13. Uzyskiwanie możliwości nadawania nazw własnym zdarzeniom (981)
- 18.14. Wykonywanie operacji atomowych wśród wątków (984)

19. Kod niezabezpieczony (987)

- 19.0. Wprowadzenie (987)
- 19.1. Kontrolowanie zmian we wskaźnikach przekazywanych do metod (988)
- 19.2. Porównywanie wskaźników (991)
- 19.3. Nawigowanie po tablicach (992)
- 19.4. Operacje na wskaźniku na tablicę stałą (994)
- 19.5. Zwracanie wskaźnika na konkretny element tablicy (995)
- 19.6. Tworzenie i używanie tablicy wskaźników (996)
- 19.7. Zamiana nieznanych typów wskaźników (998)
- 19.8. Przekształcanie ciągu znaków w char* (1000)
- 19.9. Deklarowanie struktury o stałym rozmiarze z osadzoną tablicą (1001)

20. Przybornik (1003)

- 20.0. Wprowadzenie (1003)
- 20.1. Obsługa procesów zamknięcia systemu, zarządzania mocą lub zmian w sesji użytkownika (1003)
- 20.2. Sterowanie usługą (1007)
- 20.3. Uzyskiwanie listy procesów, w których załadowano podzespół (1010)
- 20.4. Używanie kolejek komunikatów na lokalnej stacji roboczej (1012)

- 20.5. Odnajdywanie ścieżki do bieżącej wersji .NET Framework (1015)
- 20.6. Ustalanie wersji zarejestrowanych w globalnej pamięci podręcznej podzespołów (1015)
- 20.7. Odczytywanie ścieżki do katalogu Windows (1018)
- 20.8. Przechwytywanie danych wyjściowych ze standardowego strumienia wyjścia (1018)
- 20.9. Uruchamianie kodu w jego własnej domenie AppDomain (1021)
- 20.10. Ustalanie wersji systemu operacyjnego oraz pakietu Service Pack (1022)

Skorowidz (1027)