

» Idź do

- Spis treści
- Przykładowy rozdział

» 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. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2008

Excel 2007 PL. Programowanie w VBA. Vademecum profesjonalisty

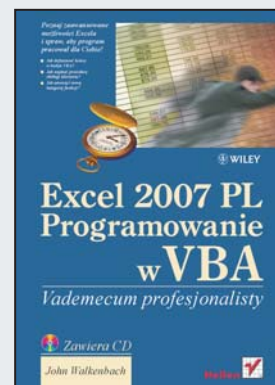
Autor: [John Walkenbach](#)

Tłumaczenie: Grzegorz Kowalczyk

ISBN: 978-83-246-2224-5

Tytuł oryginału: [Excel 2007 Power Programming with VBA \(Mr. Spreadsheet's Bookshelf\)](#)

Format: 172×245, stron: 992



Poznaj zaawansowane możliwości Excela i spraw, aby program pracował dla Ciebie!

- Jak definiować kolory w kodzie VBA?
- Jak napisać procedurę obsługi zdarzenia?
- Jak utworzyć nową kategorię funkcji?

Jeśli zajmujesz się tworzeniem aplikacji w ramach arkusza kalkulacyjnego, z którego korzystają inni użytkownicy, i zależy Ci na jak najszerzym wykorzystaniu możliwości Excela, powinieneś skorzystać z tej książki. W Excelu 2007 wprowadzono całkiem sporo nowych funkcji i mechanizmów, a dzięki informacjom zawartym w tym profesjonalnym podręczniku nauczysz się pisać programy automatyzujące wykonywanie wielu zadań w tym programie. Zdobędziesz wiedzę, która sprawi, że to on będzie pracował dla Ciebie!

Książka „Excel 2007 PL. Programowanie w VBA. Vademecum profesjonalisty” jest wręcz przepelniona procedurami, wskazówkami i pomysłami pozwalającymi na rozszerzenie funkcjonalności Excela 2007 przy użyciu języka Visual Basic for Applications. Z podręcznika, napisanego przez jeden z największych autorytetów w tej dziedzinie, dowiesz się, jak dostosowywać do własnych potrzeb formularze UserForm, kreować nowe narzędzia, wykorzystywać VBA do pracy z wykresami i tabelami przestawnymi, a także tworzyć procedury obsługi zdarzeń. Nauczysz się również pracować z procedurami i funkcjami oraz pisać programy współpracujące z innymi aplikacjami.

- Projektowanie aplikacji w programie Excel
- Język Visual Basic for Applications
- Zastosowanie formularzy UserForm
- Zaawansowane metody programowania
- Formularze i moduły klas
- Tworzenie dodatków do Excela
- Niestandardowe okna dialogowe
- Tworzenie systemów pomocy dla aplikacji
- Tabele przestawne, wykresy i obsługa zdarzeń

Zwiększ możliwości Excela i projektuj wspaniałe aplikacje!

Spis treści

O autorze	25
Przedmowa	27
Część I Podstawowe informacje	35
Rozdział 1. Skąd się wziął Excel 2007?	37
Historia arkuszy kalkulacyjnych w zarysie	37
Wszystko zaczęło się od programu VisiCalc	37
Lotus 1-2-3	38
Quattro Pro	41
Microsoft Excel	42
Excel jako dobre narzędzie dla projektantów aplikacji	47
Rola Excela w strategii Microsoftu	48
Rozdział 2. Program Excel w zarysie	49
Myślenie w kategoriach obiektów	49
Skoroszyty	50
Arkusze	50
Arkusze wykresów	52
Arkusze makr XLM	52
Arkusze dialogowe programów Excel 5 i 95	52
Interfejs użytkownika programu Excel	53
Wprowadzenie do Wstążki	54
Menu podręczne	59
Okna dialogowe	60
Skróty klawiszowe	61
Tagi inteligentne	62
Panel zadań	62
Dostosowywanie wyświetlania do własnych potrzeb	63
Wprowadzanie danych	63
Formuły, funkcje i nazwy	64
Zaznaczanie obiektów	65
Formatowanie	66
Formatowanie wartości numerycznych	66
Formatowanie wyglądu arkusza	66
Opcje ochrony	67
Ochrona formuł przed nadpisaniem	67
Ochrona struktury skoroszytu	68
Ochrona skoroszytu przy użyciu hasła	69
Ochrona kodu VBA przy użyciu hasła	69

Wykresy	70
Kształty i obiekty typu SmartArt	70
Dostęp do baz danych	71
Arkuszowe bazy danych	72
Zewnętrzne bazy danych	72
Funkcje internetowe	73
Narzędzia analizy danych	73
Konspekty	73
Dodatek Analysis ToolPak	74
Tabele przestawne	74
Dodatek Solver	74
Funkcje związane z językiem XML	75
Dodatki	75
Makra i programowanie	75
Zgodność formatu plików	75
System pomocy Excela	76
Rozdział 3. Wybrane zasady stosowania formuł	77
Formuły	77
Obliczanie formuł	78
Odwołania do komórki lub zakresu	79
Dlaczego warto używać odwołań, które nie są względne?	79
Notacja WIKI	80
Odwołania do innych arkuszy lub skoroszytów	81
Zastosowanie nazw	82
Nadawanie nazw komórkom i zakresom	83
Nadawanie nazw istniejącym odwołaniom	84
Stosowanie nazw z operatorem przecięcia	85
Nadawanie nazw kolumnom i wierszom	85
Obszar obowiązywania nazw	85
Nadawanie nazw stałym	86
Nadawanie nazw formułom	87
Nadawanie nazw obiektom	88
Błędy występujące w formułach	88
Formuły tablicowe	89
Przykład formuły tablicowej	90
Kalendarz oparty na formule tablicowej	91
Zalety i wady formuł tablicowych	92
Metody zliczania i sumowania	92
Przykłady formuł zliczających	93
Przykłady formuł sumujących	93
Inne narzędzia zliczające	93
Przetwarzanie daty i czasu	93
Wprowadzanie daty i czasu	95
Stosowanie dat sprzed roku 1900	96
Tworzenie megaformuł	97
Rozdział 4. Pliki programu Excel	99
Uruchamianie Excela	99
Formaty plików	102
Formaty plików obsługiwane w programie Excel	102
Formaty plików tekstowych	102
Formaty plików baz danych	102
Inne formaty plików	103

Praca z plikami szablonów	105
Przeglądanie dostępnych szablonów	105
Tworzenie szablonów	106
Tworzenie własnych szablonów skoroszytu	108
Budowa plików programu Excel	109
Zaglądamy do wnętrza pliku	109
Dlaczego format pliku jest taki ważny?	112
Plik QAT	113
Plik XLB	114
Pliki dodatków	114
Ustawienia Excela w rejestrze systemu Windows	115
Rejestr systemu Windows	115
Ustawienia Excela	116

Część II Projektowanie aplikacji w Excelu 119

Rozdział 5. Czym jest aplikacja arkusza kalkulacyjnego? 121

Robocza definicja aplikacji arkusza kalkulacyjnego	121
Projektant i użytkownik końcowy	123
Kim są projektanci i czym się zajmują?	123
Klasyfikacja użytkowników arkuszy kalkulacyjnych	124
Odbiorcy aplikacji arkusza kalkulacyjnego	125
Rozwiązywanie problemów przy użyciu Excela	125
Podstawowe kategorie arkuszy kalkulacyjnych	127
Arkusze robocze	127
Arkusze przeznaczone wyłącznie do użytku prywatnego	128
Aplikacje jednego użytkownika	128
Aplikacje typu „spaghetti”	128
Aplikacje użytkowe	129
Dodatki zawierające funkcje arkusza	130
Arkusze jednoblokowe	130
Modele warunkowe	130
Aplikacje bazodanowe (przechowujące dane i udzielające do nich dostępu)	131
Aplikacje komunikujące się z bazami danych	131
Aplikacje „pod klucz”	131

Rozdział 6. Podstawy projektowania aplikacji arkusza kalkulacyjnego 133

Podstawowe etapy projektowania	133
Określanie wymagań użytkownika	134
Planowanie aplikacji spełniającej wymagania użytkownika	135
Wybieranie najwłaściwszego interfejsu użytkownika	138
Dostosowywanie Wstażki do potrzeb użytkownika	139
Dostosowywanie menu podręcznego do potrzeb użytkownika	140
Tworzenie klawiszy skrótu	140
Tworzenie niestandardowych okien dialogowych	141
Zastosowanie formantów ActiveX w arkuszu	142
Rozpoczęcie prac projektowych	144
Zadania realizowane z myślą o końcowym użytkowniku	144
Testowanie aplikacji	144
Uodpornianie aplikacji na błędy popełniane przez użytkownika	146
Nadawanie aplikacji przyjaznego, intuicyjnego i estetycznego wyglądu	148
Tworzenie systemu pomocy i dokumentacji przeznaczonej dla użytkownika	149
Dokumentowanie prac projektowych	149
Przekazanie aplikacji użytkownikom	150
Aktualizacja aplikacji (kiedy to konieczne)	151

Pozostałe kwestie dotyczące projektowania	151
Wersja Excela zainstalowana przez użytkownika	151
Wersje językowe	152
Wydajność systemu	152
Tryby karty graficznej	152

Część III Język Visual Basic for Applications 155

Rozdział 7. Wprowadzenie do języka VBA 157

Podstawowe informacje o języku BASIC	157
Język VBA	158
Modele obiektowe	158
Porównanie języka VBA z językiem XLM	159
Wprowadzenie do języka VBA	159
Edytor VBE	162
Wyświetlanie karty Deweloper	163
Uruchamianie edytora VBE	163
Okna edytora VBE	164
Tajemnice okna Project Explorer	166
Dodawanie nowego modułu VBA	167
Usuwanie modułu VBA	167
Eksportowanie i importowanie obiektów	167
Tajemnice okna Code	168
Minimalizacja i maksymalizacja okien	168
Przechowywanie kodu źródłowego języka VBA	169
Wprowadzanie kodu źródłowego języka VBA	169
Dostosowywanie edytora Visual Basic	176
Karta Editor	176
Karta Editor Format	179
Karta General	180
Zastosowanie karty Docking	180
Rejestrator makr Excela	181
Co właściwie zapisuje rejestrator makr?	182
Odwołania względne czy bezwzględne?	182
Opcje związane z rejestrowaniem makr	186
Modyfikowanie zarejestrowanych makr	187
Obiekty i kolekcje	189
Hierarchia obiektów	189
Kolekcje	190
Odwoływanie się do obiektów	190
Właściwości i metody	191
Właściwości obiektów	191
Metody obiektowe	192
Tajemnice obiektu Comment	194
Pomoc dla obiektu Comment	194
Właściwości obiektu Comment	194
Metody obiektu Comment	195
Kolekcja Comments	196
Właściwość Comment	197
Obiekty zawarte w obiekcie Comment	197
Sprawdzanie, czy komórka posiada komentarz	198
Dodanie nowego obiektu Comment	199
Kilka przydatnych właściwości obiektu Application	199

Tajemnice obiektów Range	201
Właściwość Range	201
Właściwość Cells	204
Właściwość Offset	205
Co należy wiedzieć o obiektach?	207
Podstawowe zagadnienia, które należy zapamiętać	207
Dodatkowe informacje na temat obiektów i właściwości	208
Rozdział 8. Podstawy programowania w języku VBA	211
Przegląd elementów języka VBA	211
Komentarze	213
Zmienne, typy danych i stałe	214
Definiowanie typów danych	216
Deklarowanie zmiennych	216
Zasięg zmiennych	220
Zastosowanie stałych	223
Praca z łańcuchami tekstu	226
Zastosowanie dat	226
Instrukcje przypisania	227
Tablice	229
Deklarowanie tablic	229
Deklarowanie tablic wielowymiarowych	230
Deklarowanie tablic dynamicznych	230
Zmienne obiektowe	231
Typy danych definiowane przez użytkownika	232
Wbudowane funkcje VBA	233
Praca z obiektami i kolekcjami	234
Konstrukcja With ... End With	234
Konstrukcja For Each ... Next	236
Sterowanie wykonywaniem procedur	238
Polecenie GoTo	239
Konstrukcja If ... Then	239
Konstrukcja Select Case	243
Wykonywanie bloku instrukcji w ramach pętli	246
Rozdział 9. Tworzenie procedur w języku VBA	255
Kilka słów o procedurach	255
Deklarowanie procedury Sub	256
Zasięg procedury	257
Wykonywanie procedur Sub	258
Uruchamianie procedury przy użyciu polecenia Run Sub/UserForm	259
Uruchamianie procedury z poziomu okna dialogowego Makro	259
Uruchamianie procedury przy użyciu skrótu z klawiszem Ctrl	260
Uruchamianie procedury za pomocą Wstążki	261
Uruchamianie procedur za pośrednictwem niestandardowego menu podręcznego	261
Wywoływanie procedury z poziomu innej procedury	261
Uruchamianie procedury poprzez kliknięcie obiektu	265
Wykonywanie procedury po wystąpieniu określonego zdarzenia	267
Uruchamianie procedury z poziomu okna Immediate	267
Przekazywanie argumentów procedurom	268
Metody obsługi błędów	271
Przechwytywanie błędów	272
Przykłady kodu źródłowego obsługującego błędy	273

Praktyczny przykład wykorzystujący procedury Sub	275
Cel	276
Wymagania projektowe	276
Co już wiesz	276
Podejście do zagadnienia	277
Co musimy wiedzieć?	277
Wstępne rejestrowanie makr	278
Wstępne przygotowania	279
Tworzenie kodu źródłowego	280
Tworzenie procedury sortującej	281
Dodatkowe testy	285
Usuwanie problemów	286
Dostępność narzędzia	289
Ocena projektu	289
Rozdział 10. Tworzenie funkcji w języku VBA	291
Porównanie procedur Sub i Function	291
Dlaczego tworzymy funkcje niestandardowe?	292
Pierwszy przykład procedury Function	293
Funkcja niestandardowa	293
Zastosowanie funkcji w arkuszu	293
Zastosowanie funkcji w procedurze języka VBA	294
Analiza funkcji niestandardowej	294
Procedury Function	296
Deklarowanie funkcji	296
Zasięg funkcji	298
Wywoływanie procedur Function	298
Argumenty procedury Function	300
Przykłady funkcji	300
Funkcja pozbawiona argumentów	301
Funkcja z jednym argumentem	303
Funkcje z dwoma argumentami	306
Funkcja pobierająca tablicę jako argument	307
Funkcje z argumentami opcjonalnymi	308
Funkcje zwracające tablicę VBA	309
Funkcje zwracające wartość błędu	312
Funkcje o nieokreślonej liczbie argumentów	313
Emulowanie funkcji SUMA Excela	314
Wykrywanie i usuwanie błędów w funkcjach	317
Okno dialogowe Wstawianie funkcji	318
Definiowanie kategorii funkcji	319
Dodawanie opisu funkcji	320
Zastosowanie dodatków do przechowywania funkcji niestandardowych	321
Korzystanie z Windows API	322
Przykłady zastosowania funkcji interfejsu API systemu Windows	323
Identyfikacja katalogu domowego systemu Windows	323
Wykrywanie wciśnięcia klawisza Shift	324
Dodatkowe informacje na temat funkcji interfejsu API	325
Rozdział 11. Przykłady i techniki programowania w języku VBA	327
Przetwarzanie zakresów	328
Kopiowanie zakresów	328
Przenoszenie zakresów	329
Kopiowanie zakresu o zmiennej wielkości	330
Zaznaczanie różnego typu zakresów i identyfikowanie ich	330

Wprowadzanie wartości do komórki	332
Wprowadzanie wartości do następnej pustej komórki	334
Wstrzymywanie działania makra w celu umożliwienia pobrania zakresu wyznaczonego przez użytkownika	335
Zliczanie zaznaczonych komórek	336
Określanie typu zaznaczonego zakresu	337
Wydajne przetwarzanie komórek zaznaczonego zakresu przy użyciu pętli	338
Usuwanie wszystkich pustych wierszy	341
Powielanie wierszy	342
Określanie, czy zakres zawiera się w innym zakresie	344
Określanie typu danych zawartych w komórce	344
Odczytywanie i zapisywanie zakresów	345
Lepsza metoda zapisywania zakresu	346
Przenoszenie zawartości tablic jednowymiarowych	348
Przenoszenie zawartości zakresu do tablicy typu Variant	348
Zaznaczanie maksymalnej wartości zakresu	349
Kopiowanie nieciągłego zakresu komórek	350
Przetwarzanie skoroszytów i arkuszy	352
Zapisywanie wszystkich skoroszytów	352
Zapisywanie i zamykanie wszystkich skoroszytów	352
Ukrywanie wszystkich komórek arkusza poza zaznaczonym zakresem	353
Synchronizowanie arkuszy	354
Techniki programowania w języku VBA	355
Przełączanie wartości właściwości typu logicznego	355
Określanie liczby drukowanych stron	356
Wyświetlanie daty i czasu	356
Pobieranie listy czcionek	358
Sortowanie tablicy	359
Przetwarzanie grupy plików	361
Ciekawe funkcje, których możesz użyć w swoich projektach	362
Funkcja FileExists	363
Funkcja FileNameOnly	363
Funkcja PathExists	363
Funkcja RangeNameExists	364
Funkcja SheetExists	364
Funkcja WorkbookIsOpen	364
Pobieranie wartości z zamkniętego skoroszytu	365
Użyteczne, niestandardowe funkcje arkuszowe	367
Funkcje zwracające informacje o formatowaniu komórki	367
Gadający arkusz?	368
Wyświetlanie daty zapisania lub wydrukowania pliku	369
Obiekty nadrzędne	369
Zliczanie komórek, których wartości zawierają się pomiędzy dwoma wartościami	370
Zliczanie widocznych komórek zakresu	371
Wyznaczanie ostatniej niepustej komórki kolumny lub wiersza	372
Czy dany łańcuch tekstu jest zgodny z wzorcem?	373
Wydzielanie n-tego elementu łańcucha	374
Funkcja wielofunkcyjna	375
Funkcja SheetOffset	376
Zwracanie maksymalnej wartości ze wszystkich arkuszy	377
Zwracanie tablicy zawierającej unikatowe, losowo uporządkowane liczby całkowite	378
Porządkowanie zakresu w losowy sposób	379

Wywołania funkcji interfejsu Windows API	381
Określanie skojarzeń plików	381
Pobieranie informacji o napędach dyskowych	382
Pobieranie informacji dotyczących drukarki domyślnej	382
Pobieranie informacji o aktualnej rozdzielczości karty graficznej	383
Dodanie dźwięku do aplikacji	384
Odczytywanie zawartości rejestru systemu Windows i zapisywanie w nim danych	386

Część IV Praca z formularzami UserForm389

Rozdział 12. Metody tworzenia własnych okien dialogowych 391

Zanim rozpoczniesz tworzenie formularza UserForm	391
Okno wprowadzania danych	392
Funkcja InputBox języka VBA	392
Metoda InputBox Excelsa	394
Funkcja MsgBox języka VBA	396
Metoda GetOpenFilename programu Excel	399
Metoda GetSaveAsFilename programu Excel	402
Okno wybierania katalogu	403
Wybieranie katalogu przy użyciu funkcji interfejsu Windows API	403
Wybieranie katalogu przy użyciu obiektu OpenFileDialog	405
Wyświetlanie wbudowanych okien dialogowych Excelsa	406
Zastosowanie kolekcji Dialogs	407
Wykonywanie poleceń znajdujących się na Wstążce	408
Wyświetlanie formularza wprowadzania danych	408
Wyświetlanie formularza wprowadzania danych	410
Wyświetlanie formularza wprowadzania danych za pomocą VBA	410

Rozdział 13. Wprowadzenie do formularzy UserForm 411

Jak Excel obsługuje niestandardowe okna dialogowe	412
Wstawianie nowego formularza UserForm	412
Dodawanie formantów do formularza UserForm	412
Formanty okna Toolbox	414
Formant CheckBox	414
Formant ComboBox	414
Formant CommandButton	415
Formant Frame	415
Formant Image	415
Formant Label	415
Formant ListBox	415
Formant MultiPage	415
Formant OptionButton	416
Formant RefEdit	416
Formant ScrollBar	416
Formant SpinButton	416
Formant TabStrip	416
Formant TextBox	416
Formant ToggleButton	418
Modyfikowanie kontrolki formularza UserForm	418
Modyfikowanie właściwości formantów	418
Zastosowanie okna Properties	420
Wspólne właściwości	421
Zdobywanie dodatkowych informacji o właściwościach	421
Uwzględnienie wymagań użytkowników preferujących korzystanie z klawiatury	422

Wyświetlanie i zamykanie formularzy UserForm	424
Wyświetlanie formularza UserForm	424
Zamykanie formularza UserForm	425
Procedury obsługi zdarzeń	426
Przykład tworzenia formularza UserForm	427
Tworzenie formularza UserForm	427
Tworzenie kodu procedury wyświetlającej okno dialogowe	429
Testowanie okna dialogowego	430
Dodawanie procedur obsługi zdarzeń	431
Sprawdzanie poprawności danych	432
Zakończenie tworzenia okna dialogowego	433
Zdarzenia powiązane z formularzem UserForm	433
Zdobywanie informacji na temat zdarzeń	433
Zdarzenia formularza UserForm	434
Zdarzenia związane z formantem SpinButton	435
Współpraca formantu SpinButton z formantem TextBox	436
Odwoływanie się do formantów formularza UserForm	439
Dostosowywanie okna Toolbox do własnych wymagań	440
Modyfikacja ikon lub tekstu odpowiedzi	440
Dodawanie nowych zakładek	440
Dostosowywanie lub łączenie formantów	440
Dodawanie nowych formantów ActiveX	441
Tworzenie szablonów formularzy UserForm	442
Lista kontrolna tworzenia i testowania formularzy UserForm	442

Rozdział 14. Przykłady formularzy UserForm 445

Tworzenie formularza UserForm pełniącego funkcję menu	445
Zastosowanie w formularzu UserForm formantów CommandButton	446
Zastosowanie w formularzu UserForm kontrolki ListBox	446
Zaznaczanie zakresów przy użyciu formularza UserForm	447
Tworzenie okna powitalnego	449
Wyłączanie przycisku Zamknij formularza UserForm	451
Zmiana wielkości formularza UserForm	451
Powiększanie i przewijanie arkusza przy użyciu formularza UserForm	453
Zastosowania formantu ListBox	455
Formant ListBox	455
Tworzenie listy elementów formantu ListBox	456
Identyfikowanie zaznaczonego elementu listy formantu ListBox	460
Identyfikowanie wielu zaznaczonych elementów listy formantu ListBox	460
Wiele list w jednej kontrolce ListBox	461
Przenoszenie elementów listy formantu ListBox	462
Zmiana kolejności elementów listy formantu ListBox	464
Wielokolumnowe formanty ListBox	465
Zastosowanie formantu ListBox do wybierania wierszy arkusza	467
Uaktywnianie arkusza za pomocą formantu ListBox	469
Zastosowanie formantu MultiPage na formularzach UserForm	471
Korzystanie z formantów zewnętrznych	472
Animowanie etykiet	474

Rozdział 15. Zaawansowane techniki korzystania z formularzy UserForm 477

Niemodalne okna dialogowe	478
Wyświetlanie wskaźnika postępu zadania	481
Tworzenie samodzielnego wskaźnika postępu zadania	482
Wyświetlanie wskaźnika postępu zadania za pomocą formantu MultiPage	485
Wyświetlanie wskaźnika postępu zadania bez korzystania z kontrolki MultiPage	488

Tworzenie kreatorów	489
Konfigurowanie formantu MultiPage w celu utworzenia kreatora	489
Dodawanie przycisków do formularza UserForm kreatora	490
Programowanie przycisków kreatora	491
Zależności programowe w kreatorach	492
Wykonywanie zadań za pomocą kreatorów	494
Emulacja funkcji MsgBox	495
Emulacja funkcji MsgBox: kod funkcji MsgBox	495
Jak działa funkcja emulująca MsgBox?	496
Wykorzystanie funkcji MsgBox do emulacji funkcji MsgBox	498
Formularz UserForm z formantami, których położenie można zmieniać	498
Formularz UserForm bez paska tytułowego	499
Symulacja paska narzędzi za pomocą formularza UserForm	500
Formularze UserForm z możliwością zmiany rozmiaru	502
Obsługa wielu przycisków formularza UserForm za pomocą jednej procedury obsługi zdarzeń	506
Wybór koloru za pomocą formularza UserForm	508
Wyświetlanie wykresów na formularzach UserForm	510
Sposób postępowania przy wyświetlaniu wykresów na formularzach UserForm	511
Zapisywanie wykresu w postaci pliku GIF	511
Modyfikacja właściwości Picture formantu Image	511
Zaawansowane formularze danych	512
Opis ulepszonych formularzy danych	512
Instalacja dodatku — ulepszonych formularzy danych	513
Puzzle na formularzu UserForm	514

Część V Zaawansowane techniki programowania 517

Rozdział 16. Tworzenie narzędzi dla Excela w języku VBA 519

Kilka słów o narzędziach dla programu Excel	519
Zastosowanie języka VBA do tworzenia narzędzi	520
Co decyduje o przydatności narzędzia?	521
Operacje tekstowe: anatomia narzędzia	521
Podstawy tworzenia narzędzia Operacje tekstowe	522
Określenie wymagań dla narzędzia Operacje tekstowe	522
Skoroszyt narzędzia Operacje tekstowe	523
Jak działa narzędzie Operacje tekstowe?	523
Formularz UserForm dla narzędzia Operacje tekstowe	524
Moduł VBA Module1	525
Moduł formularza UserForm1	528
Poprawa wydajności narzędzia Operacje tekstowe	530
Zapisywanie ustawień narzędzia Operacje tekstowe	531
Implementacja procedury Cofnij	532
Wyświetlanie pliku pomocy	535
Umieszczanie poleceń na Wstążce	535
Ocena realizacji projektu	538
Działanie narzędzia Operacje tekstowe	539
Dodatkowe informacje na temat narzędzi Excela	539

Rozdział 17. Tabele przestawne 541

Przykład prostej tabeli przestawnej	541
Tworzenie tabel przestawnych	542
Analiza zarejestrowanego kodu tworzenia tabeli przestawnej	543
Optymalizacja wygenerowanego kodu tworzącego tabelę przestawną	544

Tworzenie złożonych tabel przestawnych	545
Dane dla złożonej tabeli przestawnej	546
Kod tworzący tabelę przestawną	546
Jak działa złożona tabela przestawna?	548
Jednoczesne tworzenie wielu tabel przestawnych	549
Tworzenie odwróconych tabel przestawnych	552
Rozdział 18. Wykresy	555
Podstawowe wiadomości o wykresach	555
Położenie wykresu	556
Rejestrator makr a wykresy	556
Model obiektu Chart	558
Popularne techniki języka VBA dotyczące wykresów	559
Tworzenie wykresów osadzonych na arkuszu danych	559
Tworzenie wykresu na arkuszu wykresu	561
Wykorzystanie VBA do uaktywnienia wykresu	561
Przenoszenie wykresu	562
Wykorzystanie VBA do deaktywacji wykresu	563
Sprawdzanie, czy wykres został uaktywniony	564
Usuwanie elementów z kolekcji ChartObjects lub Charts	564
Przetwarzanie wszystkich wykresów w pętli	565
Zmiana rozmiarów i wyrównywanie obiektów ChartObject	567
Eksportowanie wykresów	568
Eksportowanie wszystkich obiektów graficznych	569
Formatowanie wykresów za pomocą VBA	570
Formatowanie wykresu	570
Kolejne przykłady formatowania wykresów	572
Zmiana danych prezentowanych na wykresie	574
Modyfikacja danych wykresu na podstawie aktywnej komórki	574
Zastosowanie języka VBA do identyfikacji zakresu danych prezentowanych na wykresie	577
Wykorzystanie VBA do wyświetlania dowolnych etykiet danych na wykresie	580
Wyświetlanie wykresu w oknie formularza UserForm	582
Zdarzenia związane z wykresami	584
Przykład wykorzystania zdarzeń związanych z wykresami	584
Obsługa zdarzeń dla wykresów osadzonych	587
Przykład: zastosowanie zdarzeń dla wykresów osadzonych	589
Jak ułatwić sobie pracę z wykresami?	591
Drukowanie wykresów osadzonych na arkuszu	591
Wyświetlanie pokazu wykresów	591
Ukrywanie serii danych poprzez ukrywanie kolumn	592
Tworzenie wykresów, które nie są połączone z danymi	593
Wykorzystanie zdarzenia MouseOver do wyświetlania tekstu	594
Wykresy animowane	597
Przewijanie wykresów	598
Tworzenie wykresu krzywych hipocykloidalnych	600
Tworzenie wykresu zegara	601
Tworzenie wykresu interaktywnego bez użycia VBA	602
Przygotowanie danych do utworzenia wykresu interaktywnego	603
Tworzenie przycisków opcji dla interaktywnego wykresu	603
Tworzenie listy miast dla wykresu interaktywnego	604
Tworzenie zakresów danych dla wykresu interaktywnego	605
Utworzenie wykresu interaktywnego	605

Rozdział 19. Obsługa zdarzeń	607
Typy zdarzeń, które Excel może monitorować	608
Co powinienś wiedzieć o zdarzeniach	608
Sekwencje zdarzeń	609
Gdzie należy umieścić procedury obsługi zdarzeń?	609
Wyłączanie obsługi zdarzeń	610
Wprowadzanie kodu procedury obsługi zdarzeń	611
Procedury obsługi zdarzeń z argumentami	612
Zdarzenia poziomu skoroszytu	614
Zdarzenie Open	615
Zdarzenie Activate	616
Zdarzenie SheetActivate	616
Zdarzenie NewSheet	617
Zdarzenie BeforeSave	617
Zdarzenie Deactivate	617
Zdarzenie BeforePrint	618
Zdarzenie BeforeClose	619
Zdarzenia poziomu arkusza	621
Zdarzenie Change	622
Monitorowanie zmian w wybranym zakresie komórek	623
Zdarzenie SelectionChange	627
Zdarzenie BeforeDoubleClick	628
Zdarzenie BeforeRightClick	628
Zdarzenia dotyczące wykresów	629
Zdarzenia dotyczące aplikacji	630
Włączenie obsługi zdarzeń poziomu aplikacji	630
Sprawdzanie, czy skoroszyt jest otwarty	633
Monitorowanie zdarzeń poziomu aplikacji	634
Zdarzenia dotyczące formularzy UserForm	636
Zdarzenia niezwiązane z obiektami	637
Zdarzenie OnTime	637
Zdarzenie OnKey	638
 Rozdział 20. Interakcje z innymi aplikacjami	 643
Uruchamianie innych aplikacji z poziomu Excela	643
Zastosowanie funkcji Shell języka VBA	644
Zastosowanie funkcji ShellExecute interfejsu Windows API	646
Uaktywnianie aplikacji z poziomu Excela	647
Wykorzystanie instrukcji AppActivate	647
Uaktywnianie aplikacji pakietu Microsoft Office	648
Uruchamianie okien dialogowych Panelu sterowania	648
Wykorzystanie automatyzacji w programie Excel	649
Działania z obiektami innych aplikacji z wykorzystaniem automatyzacji	650
Wczesne i późne wiązanie	650
Prosty przykład późnego wiązania	653
Zarządzanie Wordem z poziomu Excela	654
Zarządzanie Excelem z poziomu innej aplikacji	657
Wysyłanie spersonalizowanych wiadomości e-mail z wykorzystaniem Outlooka	659
Wysyłanie wiadomości e-mail z załącznikami z poziomu Excela	662
Zastosowanie metody SendKeys	664
Działania z obiektami danych ActiveX (ADO)	666

Rozdział 21. Tworzenie i wykorzystanie dodatków	669
Czym są dodatki?	669
Porównanie dodatku ze standardowym skoroszytem	670
Po co tworzy się dodatki?	670
Menedżer dodatków Excela	672
Tworzenie dodatków	673
Przykład tworzenia dodatku	674
Konfiguracja skoroszytu dla przykładowego dodatku	674
Tworzenie opisu dla dodatku	675
Tworzenie dodatku	675
Instalowanie dodatku	677
Testowanie dodatków	677
Dystrybucja dodatków	678
Modyfikowanie dodatku	678
Porównanie plików XLAM i XLSM	679
Rozmiar i struktura plików XLSM i XLAM	679
Pliki XLAM — przynależność do kolekcji z poziomym VBA	679
Widoczność plików XLSM i XLAM	680
Arkusze i wykresy w plikach XLSM i XLAM	680
Dostęp do procedur VBA w dodatku	681
Przetwarzanie dodatków za pomocą kodu VBA	683
Kolekcja AddIns	683
Właściwości obiektu AddIn	685
Korzystanie z dodatku jak ze skoroszytu	687
Zdarzenia związane z obiektami AddIn	688
Optymalizacja wydajności dodatków	688
Problemy z dodatkami	689
Zapewnienie, że dodatek został zainstalowany	689
Odwoływanie się do innych plików z poziomu dodatku	691
Wykrywanie właściwej wersji Excela dla dodatku	692
 Część VI Tworzenie aplikacji	693
Rozdział 22. Tworzenie pasków narzędzi	695
Wprowadzenie do pracy ze Wstążką	695
Karty	696
VBA i Wstążka	699
Dostęp do poleceń Wstążki	700
Praca ze Wstążką	701
Aktywowanie karty	702
Dostosowywanie Wstążki do własnych potrzeb	704
Prosty przykład kodu RibbonX	704
Więcej informacji o przykładzie modyfikacji Wstążki	707
Kolejny przykład kodu RibbonX	708
Demo formantów Wstążki	712
Przykład użycia formantu DynamicMenu	717
Więcej wskazówek dotyczących modyfikacji Wstążki	720
Tworzenie pasków narzędzi w starym stylu	721
Ograniczenia funkcjonalności tradycyjnych pasków narzędzi w Excelu 2007	722
Kod tworzący pasek narzędzi	722

Rozdział 23. Praca z menu podręcznym	725
Obiekt CommandBar	725
Rodzaje obiektów CommandBar	726
Wyświetlanie menu podręcznych	726
Odwoływanie się do elementów kolekcji CommandBars	727
Odwołania do formantów w obiekcie CommandBar	728
Właściwości formantów obiektu CommandBar	729
Wyświetlanie wszystkich elementów menu podręcznego	730
Wykorzystanie VBA do dostosowywania menu podręcznego	731
Resetowanie menu podręcznego	731
Wyłączanie menu podręcznego	732
Wyłączanie wybranych elementów menu podręcznego	733
Dodawanie nowego elementu do menu podręcznego Cell	733
Dodawanie nowego podmenu do menu podręcznego	735
Menu podręczne i zdarzenia	737
Automatyczne tworzenie i usuwanie menu podręcznego	737
Wyłączanie lub ukrywanie elementów menu podręcznego	738
Tworzenie kontekstowych menu podręcznych	739
Rozdział 24. Tworzenie systemów pomocy w aplikacjach	743
Dlaczego należy tworzyć systemy pomocy w aplikacjach?	743
Systemy pomocy wykorzystujące komponenty Excela	744
Wykorzystanie komentarzy do tworzenia systemów pomocy	746
Wykorzystanie pól tekstowych do wyświetlania pomocy	747
Wykorzystanie arkusza do wyświetlania tekstu pomocy	748
Wyświetlanie pomocy w oknie formularza UserForm	749
Wyświetlanie pomocy w oknie przeglądarki sieciowej	752
Zastosowanie plików w formacie HTML	752
Zastosowanie plików w formacie MHTML	753
Wykorzystanie systemu HTML Help	754
Wykorzystanie metody Help do wyświetlania pomocy w formacie HTML Help	756
Zastosowanie funkcji API do wyświetlania pomocy w formacie HTML Help	757
Powiązanie pliku pomocy z aplikacją	757
Przypisanie tematów pomocy do funkcji VBA	758
Rozdział 25. Tworzenie aplikacji przyjaznych dla użytkownika	761
Czym jest aplikacja przyjazna dla użytkownika?	761
Kreator amortyzacji pożyczek	762
Obsługa Kreatora amortyzacji pożyczek	762
Struktura skoroszytu Kreatora amortyzacji pożyczek	764
Jak działa Kreator amortyzacji pożyczek?	764
Potencjalne usprawnienia Kreatora amortyzacji pożyczek	771
Wskazówki dotyczące projektowania aplikacji	771
Część VII Inne zagadnienia	773
Rozdział 26. Problem kompatybilności aplikacji	775
Co to jest kompatybilność?	775
Rodzaje problemów ze zgodnością	776
Unikaj używania nowych funkcji i mechanizmów	777
Czy aplikacja będzie działać na komputerach Macintosh?	778
Tworzenie aplikacji dla wielu wersji narodowych	779
Aplikacje obsługujące wiele języków	781
Obsługa języka w kodzie VBA	781

Wykorzystanie właściwości lokalnych	781
Identyfikacja ustawień systemu	782
Ustawienia daty i godziny	785
Rozdział 27. Operacje na plikach wykonywane za pomocą kodu VBA	787
Najczęściej wykonywane operacje na plikach	788
Zastosowanie poleceń języka VBA do wykonywania operacji na plikach	788
Zastosowanie obiektu FileSystemObject	792
Wyświetlanie rozszerzonych informacji o plikach	795
Operacje z plikami tekstowymi	797
Otwieranie plików tekstowych	798
Odczytywanie plików tekstowych	799
Zapisywanie danych do plików tekstowych	799
Przydzielanie numeru pliku	799
Określanie lub ustawianie pozycji w pliku	800
Instrukcje pozwalające na odczytywanie i zapisywanie plików	800
Przykłady wykonywania operacji na plikach	801
Importowanie danych z pliku tekstowego	801
Eksportowanie zakresu do pliku tekstowego	801
Importowanie pliku tekstowego do zakresu	803
Rejestrowanie wykorzystania Excela	804
Filtrowanie zawartości pliku tekstowego	804
Eksportowanie zakresu komórek do pliku HTML	805
Eksportowanie zakresu komórek do pliku XLM	807
Pakowanie i rozpakowywanie plików	809
Pakowanie plików do formatu ZIP	810
Rozpakowywanie plików ZIP	812
Rozdział 28. Operacje na składnikach języka VBA	815
Podstawowe informacje o środowisku IDE	815
Model obiektowy środowiska IDE	816
Kolekcja VBProjects	818
Wyświetlanie wszystkich składników projektu VBA	819
Wyświetlanie wszystkich procedur VBA w arkuszu	821
Zastępowanie modułu uaktualnioną wersją	822
Zastosowanie języka VBA do generowania kodu VBA	824
Zastosowanie VBA do umieszczenia formantów na formularzu UserForm	825
Operacje z formularzami UserForm w fazie projektowania i wykonania	826
Dodanie 100 przycisków CommandButton w fazie projektowania	827
Programowe tworzenie formularzy UserForm	829
Prosty przykład formularza UserForm	829
Użyteczny (ale już nie tak prosty) przykład dynamicznego formularza UserForm	831
Rozdział 29. Moduły klas	837
Czym jest moduł klasy?	837
Przykład: utworzenie klasy NumLock	838
Wstawianie modułu klasy	839
Dodawanie kodu VBA do modułu klasy	839
Wykorzystanie klasy NumLock	842
Dodatkowe informacje na temat modułów klas	843
Nadawanie nazwy klasie obiektów	843
Programowanie właściwości obiektów	843
Programowanie metod obiektów	845
Zdarzenia definiowane w module klasy	845

Przykład: klasa CSVFileClass	846
Zmienne poziomu modułu dla klasy CSVFileClass	846
Definicje właściwości klasy CSVFileClass	846
Definicje metod klasy CSVFileClass	847
Wykorzystanie obiektów CSVFileClass	848
Rozdział 30. Praca z kolorami	851
Definiowanie kolorów	852
Model kolorów RGB	852
Model kolorów HSL	852
Konwersja kolorów	854
Skala szarości	855
Zamiana kolorów na skalę szarości	856
Wyświetlanie wykresów w skali szarości	857
Eksperymenty z kolorami	858
Praca z motywami dokumentów Excela 2007	859
Kilka słów o motywach dokumentów	859
Kolory motywów dokumentów	859
Wyświetlanie wszystkich kolorów motywu	862
Praca z obiektami Shape	865
Kolor tła kształtu	865
Inne rodzaje wypełnień kształtów	867
Więcej informacji o kształtach	871
Modyfikacja kolorów wykresów	872
Rozdział 31. Często zadawane pytania na temat programowania w Excelu	875
Ogólne pytania dotyczące programu Excel	876
Jak zarejestrować makro?	876
Jak uruchomić makro?	876
Co mam zrobić, jeżeli karta Deweloper nie jest widoczna?	876
Zarejestrowałem nowe makro i zapisałem skoroszyt na dysku. Po ponownym otwarciu skoroszytu okazało się, że wszystkie makra zniknęły! Co się z nimi stało?	876
Przed zapisaniem skoroszytu w formacie XLSM oznaczyłem jako komentarze wszystkie polecenia VBA w module i chciałem wrócić do nich później. Kiedy ponownie otwarłem skoroszyt, okazało się, że cały kod VBA został usunięty. Co się stało?	877
Jak ukryć Wstążkę, aby nie zajmowała na ekranie tyle miejsca?	877
Gdzie podziały się moje stare paski narzędzi?	877
Czy mogę nadal korzystać z moich starych, „pływających” pasków narzędzi?	877
Gdzie mogę znaleźć przykłady procedur VBA?	877
Jak mogę w Excelu 2007 ukryć pasek stanu?	877
Czy istnieje jakieś narzędzie, które potrafi zamienić moją aplikację programu Excel w samodzielny, wykonywalny plik EXE?	878
Dlaczego naciśnięcie kombinacji klawiszy Ctrl+A nie zaznacza wszystkich komórek mojego arkusza?	878
Dlaczego polecenie Widoki niestand. nie jest aktywne?	878
Jak umieścić listę rozwijaną w komórce, tak aby użytkownik mógł wybierać odpowiednie wartości z listy?	878
Czy mogę użyć opisanej powyżej metody tworzenia listy rozwijanej, jeżeli elementy listy są przechowywane na innym arkuszu lub w innym skoroszycie?	878
Używam właściwości Application.Calculation do ustawienia trybu przeliczania arkusza na ręczny. Wygląda jednak na to, że zmiana ma zastosowanie do wszystkich skoroszytów, a nie tylko do skoroszytu aktywnego	879

Dlaczego naciśnięcie klawisza F4 nie powoduje powtórzenia niektórych operacji? ...	879
Co się stało z możliwością odczytywania na głos zawartości komórek?	879
Jak mogę zwiększyć liczbę kolumn arkusza?	879
Jak mogę zwiększyć liczbę wierszy arkusza?	880
Otworzyłem skoroszyt i okazało się, że arkusze mają tylko po 65 546 wierszy. Co się stało?	880
Jak mogę w starych skoroszytach używać nowych czcionek?	880
Jak wyświetlić podgląd wydruku?	880
Kiedy przełączam się na szablon nowego dokumentu, mój arkusz nie mieści się już dłużej na jednej stronie	880
Jak pozbyć się irytującego, kropkowanego oznaczenia podziałów strony po przełączeniu do widoku normalnego?	881
Czy mogę dodać polecenie Pokaż podziały stron do paska narzędzi Szybki dostęp?	881
Zmieniłem orientację tekstu w komórce na pionową (karta Narzędzia główne, grupa opcji Wyrównanie). Jak przywrócić normalny układ tekstu w komórce? W menu podręcznym polecenia Orientacja nie ma opcji Tekst poziomy	881
Próbuję nadać tabeli nowy styl formatowania, ale nie przynosi to żadnego efektu. Co mogę zrobić w takiej sytuacji?	881
Co powinienem zrobić, aby programy mojego pakietu Office 2007 mogły zapisywać dokumenty w formacie PDF?	881
Czy mogę zmienić kolor karty arkusza?	881
Czy mogę zmienić czcionkę, która jest używana do wyświetlania nazwy arkusza na karcie?	882
Czy mogę zmienić domyślną czcionkę i kolor komentarza komórki?	882
Czy mogę utworzyć makro VBA, które odtwarza dźwięki?	882
Kiedy otwieram skoroszyt, Excel pyta, czy chcę zaktualizować łącza. Przeszukałem wszystkie formuły i nie mogę znaleźć żadnego łącza w tym skoroszycie. Czy to błąd Excela?	882
Dlaczego Excel przy każdym uruchomieniu zawiesza się?	883
Pytania dotyczące edytora Visual Basic	883
Czy można wykorzystać rejestrator makr VBA do zarejestrowania wszystkich makr?	883
Włączyłem rejestrowanie makr podczas edytowania wykresu, ale wiele wykonywanych poleceń nie zostało zarejestrowanych	883
Zdefiniowałem kilka ogólnych makr. Chciałbym, aby były dostępne przez cały czas. W jaki sposób najlepiej to zrobić?	884
Nie mogę odnaleźć osobistego arkusza makr. Gdzie on się podział?	884
Zablokowałem dostęp do projektu VBA za pomocą hasła, a potem je zapomniałem. Czy istnieje sposób odblokowania projektu?	884
W jaki sposób napisać makro zmieniające hasło projektu?	884
Kiedy wstawiam nowy moduł, zawsze rozpoczyna się on od wiersza Option Explicit. Co to znaczy?	884
Dlaczego kod VBA jest wyświetlany w różnych kolorach? Czy mogę je zmienić?	884
Chcę usunąć moduł VBA za pomocą kodu. Czy mogę to zrobić?	885
Napisałem w Excelu 2000 makro, które dodaje kod VBA do projektu VB. Kiedy uruchamiam je w Excelu 2003, wyświetla się komunikat o błędzie. W czym problem?	885
W jaki sposób napisać makro zmieniające zabezpieczenia makr ustawione przez użytkownika? Chciałbym uniknąć wyświetlania komunikatu „Ten skoroszyt zawiera makra” w momencie otwierania mojej aplikacji	885
Nie rozumiem, jak działa opcja UserInterfaceOnly w kontekście zabezpieczania arkusza	885
W jaki sposób można stwierdzić, czy skoroszyt zawiera wirusa w makrze?	886
Mam problem z operatorem konkatenacji (&) w VBA. Podczas próby połączenia dwóch łańcuchów znaków wyświetlany jest komunikat o błędzie	886

Wydaje mi się, że operator kontynuacji wiersza w języku VBA (znak podkreślenia) nie działa	886
Z mojej aplikacji Excela korzysta wielu użytkowników. Na niektórych komputerach nie działają procedury obsługi błędów VBA. Dlaczego?	886
Pytania dotyczące procedur	886
Jaka jest różnica pomiędzy procedurą VBA a makrem?	886
Czym jest procedura?	886
Czym jest typ danych Variant?	887
Jaka jest różnica pomiędzy tablicą typu Variant a tablicą zawierającą dane typu Variant?	887
Co to jest znak definicji typu?	887
Chciałbym utworzyć procedurę, która automatycznie modyfikuje formatowanie komórki na podstawie wprowadzonych danych. Jeżeli na przykład wprowadzę wartość większą od zera, tło komórki ma być czerwone. Czy to możliwe?	888
Funkcja formatowania warunkowego jest przydatna, ale chciałbym również wykonać inne działania w czasie, kiedy w komórce są wprowadzane dane	888
Jakie inne rodzaje zdarzeń można monitorować?	888
Próbowałem zdefiniować procedurę obsługi zdarzenia (Sub Workbook_Open), ale procedura nie wykonuje się podczas otwierania skoroszytu. W czym problem?	888
Wiem, że można napisać procedurę obsługi zdarzenia dla określonego skoroszytu, ale czy można napisać procedurę obsługi zdarzenia działającą dla dowolnego otwartego skoroszytu?	888
Potrafię tworzyć formuły w Excelu. Czy w języku VBA są wykorzystywane te same operatory matematyczne i logiczne?	889
W jaki sposób można uruchomić procedurę znajdującą się w innym skoroszytcie?	889
Za pomocą języka VBA utworzyłem kilka funkcji. Chciałbym wykorzystać je w formuлах arkusza, ale poprzedzanie nazwy funkcji nazwą arkusza jest dla mnie niewygodne. Czy jest jakiś sposób obejścia tego problemu?	889
Chciałbym, aby pewien skoroszyt był ładowany za każdym razem, kiedy uruchamiam Excela. Chciałbym też, aby makro zapisane w tym skoroszytcie było wykonywane automatycznie. Czy wymagam zbyt wiele?	890
Mam skoroszyt, w którym zdefiniowałem procedurę Workbook_Open. Czy jest sposób zabezpieczenia się przed wykonywaniem tej procedury w momencie otwierania skoroszytu?	890
Czy z poziomu procedury VBA można uzyskać dostęp do wartości komórki arkusza, który nie jest otwarty?	890
Jak uniknąć wyświetlania pytania „Czy chcesz zapisać plik...” w momencie zamykania skoroszytu z poziomu VBA?	890
Co zrobić, aby makro uruchamiało się co godzinę?	891
W jaki sposób nie dopuścić do wyświetlania makra na liście makr?	891
Czy można zapisać wykres w formacie GIF?	891
Czy zmienne zadeklarowane w procedurze VBA są dostępne w innych procedurach VBA? A jak wygląda sytuacja w przypadku, kiedy procedurę zdefiniowano w innym module lub w innym skoroszytcie?	891
Pytania dotyczące funkcji	892
Utworzyłem funkcję VBA, która będzie używana w formuлах jako funkcja arkuszowa. Niestety próba jej użycia zawsze kończy się błędem #NAZWA?.	
Co jest nie tak?	892
Napisałem funkcję VBA, która działa bez zarzutu, jeżeli wywołam ją z innej procedury. Funkcja nie działa jednak, gdy zostanie użyta w formule.	
W czym problem?	892

Utworzyłem funkcję arkusza. Kiedy próbuję z niej skorzystać za pomocą okna dialogowego Wstawianie funkcji, wyświetla się komunikat „Pomoc niedostępna”. Co zrobić, aby w oknie Wstawianie funkcji wyświetlał się opis mojej funkcji?	892
Czy w oknie dialogowym Wstawianie funkcji można również wyświetlić pomoc na temat argumentów zdefiniowanej funkcji?	892
Zdefiniowana przeze mnie funkcja wyświetla się w kategorii Użytkownika okna dialogowego Wstawianie funkcji.	
W jaki sposób spowodować, aby wyświetlała się w innej kategorii?	893
Jak utworzyć nową kategorię funkcji?	893
Zdefiniowałem funkcję do wykorzystania w formułach arkusza. Co należy zrobić, aby funkcja zwracała wartość błędu (#ARG!), jeżeli użytkownik wprowadzi niepoprawne argumenty?	893
Jak wymusić przeliczenie formuł, w których wykorzystano funkcję użytkownika?	894
Czy można użyć wbudowanych funkcji Excela w kodzie VBA?	894
Czy jest sposób, aby wymusić koniec wiersza w wyświetlanym oknie informacyjnym MsgBox?	894
Pytania dotyczące obiektów, właściwości, metod i zdarzeń	894
Nie rozumiem pojęcia obiektów. Czy jest gdzieś lista obiektów Excela, które można wykorzystać?	894
Przeraża mnie liczba dostępnych właściwości i metod. Jak mogę się dowiedzieć, jakie metody i właściwości są dostępne dla określonego obiektu?	895
Czym są kolekcje? Czy są to obiekty?	895
Kiedy próbuję odwołać się do arkusza w kodzie VBA, uzyskuję komunikat o błędzie „subscript out of range” (indeks poza zakresem). Ja przecież nie używam żadnych indeksów. O co tu chodzi?	895
Jak zablokować możliwość przewijania arkusza przez użytkownika?	896
Czym się różni metoda Select obiektu Range od metody Goto obiektu Application?	896
Jaka jest różnica pomiędzy uaktywnieniem zakresu a jego zaznaczeniem?	896
Czy istnieje łatwy sposób usunięcia z arkusza wszystkich wartości i pozostawienia formuł bez zmian?	896
Potrafię napisać kod VBA, który zaznacza zakres na podstawie podanego adresu komórki, ale w jaki sposób napisać kod, który zaznacza zakres tylko na podstawie numeru wiersza i kolumny?	897
Czy jakieś polecenie VBA powoduje zakończenie pracy Excela? Kiedy próbuję zarejestrować polecenie Plik/Zakończ, Excel kończy działanie, zanim zdolał obejrzeć zarejestrowany kod	897
Jak wyłączyć aktualizowanie ekranu w czasie działania makra?	897
Jak najłatwiej zdefiniować nazwę zakresu za pomocą kodu VBA?	897
Jak sprawdzić, czy komórce lub zakresowi nadano nazwę?	898
Czy można wyłączyć przyciski Ustawienia i Marginesy wyświetlane w oknie Podgląd wydruku Excela?	898
Czy można wyświetlać komunikaty na pasku stanu w czasie działania makra? Napisałem makro, które długo działa. Byłoby dobrze, gdyby postęp wykonania wyświetlał się na pasku stanu	898
Zarejestrowałem makro VBA kopiujące zakres i wklejające go do innego obszaru. Wykorzystałem w nim metodę Select. Czy istnieje lepszy sposób od wycinania i wklejania?	898
Nie udało mi się znaleźć metody pozwalającej na sortowanie tablicy VBA. Czy to oznacza, że muszę skopiować wartości do arkusza i zastosować metodę Range.Sort?	899

Moje makro działa z zaznaczonymi komórkami, ale nie działa w przypadku zaznaczenia innych elementów (np. wykresu). W jaki sposób sprawdzić, czy zaznaczono zakres komórek?	899
Jak sprawdzić, czy uaktywniono wykres?	899
Chcę, by makro VBA zliczało liczbę wierszy zaznaczonych przez użytkownika. Zastosowanie metody Selection.Rows.Count nie działa, jeżeli zostaną zaznaczone wiersze, które ze sobą nie sąsiadują. Czy to jest błąd?	899
Korzystam z Excela do tworzenia faktur. Czy istnieją sposoby generowania niepowtarzalnych numerów faktur?	900
Czy istnieje właściwość skoroszytu, która uniemożliwia ukrycie okna Excela przez okno innej aplikacji?	900
Czy istnieje instrukcja VBA umożliwiająca zaznaczenie ostatniej pozycji w kolumnie lub wierszu? Ręcznie można to zrobić za pomocą klawiszy Ctrl+Shift+strzałka w dół lub Ctrl+Shift+strzałka w prawo, ale jak zrobić to samo za pomocą makra?	900
Jak uzyskać informację o ostatniej niepustej komórce w podanej kolumnie?	901
Odwolań w języku VBA bywają bardzo długie, zwłaszcza jeżeli trzeba użyć pełnej specyfikacji arkusza i skoroszytu. Czy istnieje sposób skrócenia tych odwołań?	901
Czy można zadeklarować tablicę, jeżeli nie wiadomo, ile elementów ma ona zawierać?	901
Czy mogę umożliwić użytkownikowi cofnięcie skutków działania makra?	902
Czy mogę zatrzymać działanie makra, tak aby użytkownik mógł wprowadzić dane do wybranej komórki?	902
W języku VBA istnieje funkcja InputBox, ale jest także metoda InputBox obiektu Application. Czy to jest to samo?	902
Aby napisać instrukcję VBA tworzącą formułę, muszę użyć znaku cudzysłowu wewnątrz łańcucha znaków ujętego w cudzysłów. Jak to zrobić?	902
Utworzyłem tablicę, ale pierwszy jej element jest traktowany jak drugi. W czym problem?	903
Chciałbym, aby kod VBA wykonywał się tak szybko, jak to tylko możliwe. Czy są na to jakieś sposoby?	903
Pytania dotyczące formularzy UserForm	904
Chcę pobrać od użytkownika zaledwie kilka informacji. Wydaje mi się, że zastosowanie formularza UserForm to zbyt dużo. Czy istnieje jakieś alternatywne rozwiązanie?	904
W formularzu UserForm zdefiniowałem 12 przycisków. Jak przypisać pojedyncze makro, które będzie wykonywane w momencie kliknięcia dowolnego z nich?	904
Jak wyświetlić wykres w formularzu UserForm?	904
Jak usunąć przycisk „X” z paska tytułu formularza UserForm? Nie chcę, aby użytkownik mógł zamykać okno w ten sposób	904
Utworzyłem formularz UserForm, którego formanty są powiązane z arkuszem za pomocą właściwości ControlSource. Czy jest to najlepszy sposób rozwiązywania tego problemu?	905
Czy można utworzyć tablicę formantów dla formularza UserForm? Można to zrobić w Visual Basicu, ale nie wiem, jak się to robi w języku VBA	905
Czy jest różnica pomiędzy ukryciem formularza UserForm a usunięciem go z pamięci?	905
Co zrobić, aby formularz UserForm pozostał otwarty podczas wykonywania innych działań?	905
W Excelu 97 powstaje błąd kompilacji przy próbie skorzystania z instrukcji UserForm1.Show vbModeless. Czy można napisać kod, który powoduje, że okno jest niemodalne w Excelu 2000 i wersjach nowszych, ale pozostaje modalne w Excelu 97?	906

Chciałbym wyświetlić wskaźnik postępu zadania dla długo wykonujących się procesów, podobny do tych, które wyświetlają się w programach instalujących oprogramowanie. Jak to zrobić?	906
Czy można wykorzystać narzędzie rysowania Excela do utworzenia prostych rysunków w formularzu UserForm?	906
Jak wygenerować listę plików i katalogów w formularzu UserForm, aby użytkownik mógł wybrać plik z listy?	907
Chciałbym połączyć łańcuchy znaków i wyświetlić je za pomocą kontrolki ListBox. Kiedy jednak to robię, nie są one właściwie wyrównane. Co zrobić, aby pomiędzy łańcuchami znaków były wyświetlane równe odstępy?	907
Czy istnieje prosty sposób wypełnienia pola listy lub pola kombi?	907
Czy można wyświetlać wbudowane okna dialogowe Excela za pomocą kodu VBA?	907
Wypróbowałem technikę opisaną w odpowiedzi na poprzednie pytanie i uzyskałem komunikat o błędzie. Dlaczego?	908
Za każdym razem, kiedy tworzę formularze UserForm, dodaję przyciski OK i Anuluj. Czy jest jakiś sposób, aby kontrolki te wyświetlały się automatycznie?	908
Czy można utworzyć formularz UserForm bez paska tytułu?	908
Kiedy kliknę przycisk w zdefiniowanym przeze mnie formularzu UserForm, nic się nie dzieje. Czy robię coś źle?	908
Czy można utworzyć formularz UserForm, którego rozmiar jest taki sam, niezależnie od rozdzielczości ekranu?	908
Czy można utworzyć pole w formularzu UserForm umożliwiające określenie zakresu arkusza poprzez jego wskazanie?	908
Czy można zmienić początkową pozycję formularza UserForm?	908
Czy można utworzyć formularz UserForm tak, aby użytkownik mógł zmieniać jego rozmiar?	909
Pytania dotyczące dodatków	909
Gdzie można znaleźć dodatki Excela?	909
W jaki sposób zainstalować dodatek?	909
Kiedy instaluję dodatek z okna dialogowego Dodatki, wyświetla się on bez nazwy i opisu. Jak dodać opis dodatku?	909
Mam kilka dodatków, których nie używam, ale nie wiem, w jaki sposób usunąć je z listy Dostępne dodatki w oknie dialogowym Dodatki	910
Jak tworzy się dodatki?	910
Próbowałem utworzyć dodatek, ale w polu Zapisz jako typ nie ma możliwości wyboru dodatku	910
Czy powinienem przekształcić wszystkie moje ważne skoroszyty na dodatki?	910
Czy powinienem zachować dwie kopie skoroszytu: wersję XLSM i XLAM?	910
Jak modyfikować dodatek po jego utworzeniu?	910
Czym się różni plik XLSM od pliku XLAM utworzonego na podstawie tego pliku XLSM? Czy plik XLAM to wersja skompilowana? Czy działa szybciej?	911
Czy można zablokować możliwość przeglądania kodu dodatku przez innych użytkowników?	911
Czy dodatki XLAM są bezpieczne? Mówiąc inaczej, jeżeli dostarczę użytkownikom plik XLAM, czy mogą mieć pewność, że nikt nie będzie przeglądał kodu?	911
Pytania dotyczące pasków poleceń	911
Jak użyć VBA do umieszczenia nowego przycisku polecenia na Wstążce?	911
Jak dodać makro do paska narzędzi Szybki dostęp?	911
Umieściłem na pasku narzędzi Szybki dostęp przycisk makra, ale jego kliknięcie powoduje błąd	912
Jak użyć VBA do aktywowania określonej karty Wstążki?	912
Jak zablokować możliwość wyświetlania menu podręcznego po kliknięciu prawym przyciskiem myszy?	912

Dodatki	913
Dodatek A Zasoby online dotyczące Excela	915
Pomoc systemowa programu Excel	915
Pomoc techniczna firmy Microsoft	916
Opcje pomocy technicznej	916
Baza wiedzy firmy Microsoft	916
Strona domowa programu Microsoft Excel	916
Strona domowa pakietu Microsoft Office	916
Internetowe grupy dyskusyjne	917
Dostęp do grup dyskusyjnych za pomocą czytników grup dyskusyjnych	917
Dostęp do grup dyskusyjnych za pomocą przeglądarki sieciowej	917
Wyszukiwanie informacji w grupach dyskusyjnych	918
Strony internetowe WWW	919
Strona domowa Spreadsheet	919
Strona Daily Dose of Excel	920
Strona o Excelu Jona Peltiera	920
Pearson Software Consulting	920
Strona o Excelu Stephena Bullena	920
Strony o Excelu Davida McRitchie	920
Mr. Excel	920
Dodatek B Instrukcje i funkcje VBA	921
Wywoływanie funkcji Excela w instrukcjach VBA	924
Dodatek C Kody błędów VBA	931
Dodatek D Zawartość płyty CD-ROM	935
Wymagania systemowe	935
Korzystanie z płyty CD-ROM	935
Co znajduje się na płycie CD-ROM	936
Przykładowe skoroszyty omawiane w naszej książce	936
Rozwiązywanie problemów	950
Skorowidz	951

Rozdział 4.

Pliki programu Excel

W tym rozdziale:

- Różne sposoby uruchamiania programu Excel
- Formaty plików, które Excel może otwierać i zapisywać
- Wprowadzenie do nowego formatu XML plików Excels 2007
- Jak Excel 2007 korzysta z rejestrów systemu Windows

Jeżeli zamierzasz używać Excels do realizacji różnych zaawansowanych zadań, powinieneś zapoznać się z jego specyfiką działania i zrozumieć, co dzieje się podczas jego uruchamiania. Warto również poznać formaty plików obsługiwane i tworzone przez Excels.

Uruchamianie Excels

Excel może być uruchamiany na różne sposoby w zależności od tego, jak został zainstalowany. Sposoby uruchamiania obejmują kliknięcie ikony programu znajdującej się na pulpicie, użycie przycisku *Start* i dwukrotne kliknięcie lewym przyciskiem myszy pliku powiązanego z Excels (na przykład pliku skoroszytu). Wszystkie metody ostatecznie powodują załadowanie pliku wykonywalnego o nazwie *excel.exe*.

Podczas uruchamiania Excel wykonuje następujące operacje:

- Odczytuje ustawienia przechowywane w rejestrze systemu Windows.
- Otwiera plik **.xlb*, w którym przechowywane są informacje o modyfikacjach menu i pasków narzędzi.
- Otwiera wszystkie zainstalowane dodatki (czyli takie, które zostały uaktywnione w oknie dialogowym *Dodatki*).
- Otwiera wszystkie skoroszyty przechowywane w folderze *XLStart*.
- Otwiera wszystkie skoroszyty przechowywane w alternatywnym folderze startowym (zdefiniowanym na karcie *Zaawansowane* okna dialogowego *Opcje programu Excel*).
- Tworzy nowy, pusty skoroszyt — o ile wcześniej nie zostały otwarte inne skoroszyty zdefiniowane przez użytkownika bądź przechowywane w folderze *XLStart* lub alternatywnym folderze startowym.



Aby zmienić domyślne formaty (lub zawartość) tworzonych nowych skoroszytów, powinieneś utworzyć domyślny skoroszyt, a następnie zapisać go jako szablon w katalogu *XLStart* pod nazwą *Skoroszyt.xltx*. Szczegóły dotyczące tworzenia i używania plików szablonów znajdziesz w systemie pomocy Excela.

Excel może być zainstalowany w dowolnym katalogu, ale w większości przypadków plik wykonywalny Excela możesz znaleźć w domyślnym katalogu instalacyjnym:

C:\Program Files\Microsoft Office\Office12\EXCEL.EXE

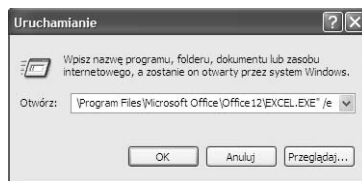
Możesz utworzyć jeden lub kilka skrótów do tego pliku, a każdy ze skrótów może wykorzystywać różne zestawy parametrów i przełączników wiersza poleceń wykorzystywanych do uruchamiania Excela. Zestawienie przełączników stosowanych w wierszu poleceń zamieszczono w tabeli 4.1.

Tabela 4.1. *Przełączniki wiersza poleceń Excela*

Przełącznik	Opis działania
<i>nazwa_pliku</i>	Otwiera plik o podanej nazwie. Nazwa pliku jest parametrem uruchamiania Excela i nie wymaga podawania żadnego przełącznika.
<i>/r nazwa_pliku</i>	Otwiera plik o podanej nazwie w trybie tylko do odczytu.
<i>/t nazwa_pliku</i>	Otwiera plik o podanej nazwie jako plik szablonu.
<i>/n nazwa_pliku</i>	Otwiera plik o podanej nazwie jako plik szablonu (tak samo jak w przypadku przełącznika <i>/t</i>).
<i>/e</i>	Uruchamia Excela bez tworzenia nowego skoroszytu i bez wyświetlania ekranu powitalnego.
<i>/p katalog</i>	Ustawia aktywną ścieżkę na katalog inny niż katalog domyślny.
<i>/s</i>	Uruchamia Excela w trybie awaryjnym, w którym nie są ładowane żadne dodatki ani skoroszyty przechowywane w folderze <i>XLStart</i> i alternatywnym folderze startowym.
<i>/embedded</i>	Uruchamia niewidoczną instancję Excela (niezalecane).
<i>/m</i>	Zmusza Excela do utworzenia nowego skoroszytu zawierającego pojedynczy arkusz makr aplikacji Microsoft Excel 4.0 (opcja przestarzała).

Z przełącznikami Excela możesz eksperymentować po wybraniu w systemie Windows polecenia *Start/Uruchom*. Ścieżkę do programu Excel powinieneś umieścić w znakach cudzysłowu, a na końcu, po spacji, dodać odpowiednie przełączniki. Przykład przedstawiono na rysunku 4.1.

Rysunek 4.1.
*Uruchamianie Excela
za pomocą okna
dialogowego Uruchom*



Jeden ze sposobów określenia przełącznika polega na poddaniu edycji właściwości skrótu uruchamiającego program. Jeśli na przykład chcesz, aby katalogiem domyślnym został folder *c:\ExcelPliki*, powinieneś odpowiednio zmodyfikować skrót uruchamiający Excela i użyć przełącznika */p*.



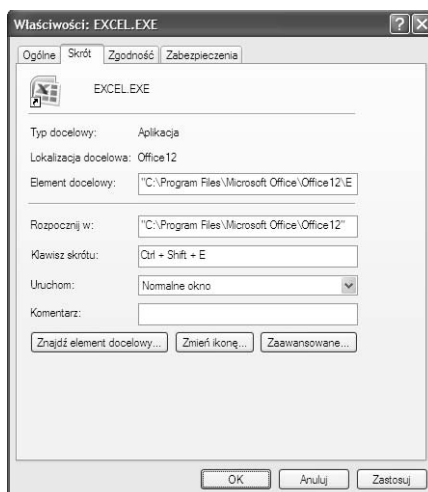
Poniżej podano sposób modyfikowania skrótu dla systemu Windows XP.

Aby to zrobić, odszukaj ikonę, której kliknięcie uruchamia Excela, kliknij ją prawym przyciskiem myszy i z menu podręcznego wybierz polecenie *Właściwości*. Na ekranie pojawi się okno dialogowe właściwości skrótu. Kliknij kartę *Skrót* i zmień zawartość pola *Element docelowy*, dodając odpowiedni przełącznik (patrz rysunek 4.2); zawartość pola *Element docelowy* można zmienić na przykład na następującą:

"C:\Program Files\Microsoft Office\Office12\EXCEL.EXE" /p C:\PlikiX1

Rysunek 4.2.

Dostosowywanie skrótu uruchamiającego program Excel



Pamiętaj, że ścieżka do pliku *EXCEL.EXE* może być inna w przypadku różnych wersji Excela i sposobów instalacji.

Do uruchamiania Excela możesz również zdefiniować odpowiedni skrót klawiszowy, co w praktyce może się okazać bardzo użyteczne. Naciśnięcie takiego skrótu w sytuacji, kiedy Excel jest już uruchomiony, powoduje aktywowanie okna programu.



W jednym systemie można uruchomić wiele kopii Excela. Każda kopia jest traktowana jak oddzielne zadanie. Wielu użytkowników z powodzeniem używa nawet różnych *wersji* Excela zainstalowanych na jednym komputerze. Aby osiągnąć jak najlepsze rezultaty, kolejne wersje należy instalować zgodnie z kolejnością ich pojawiania się na rynku.

Formaty plików

Co prawda domyślnym typem pliku programu Excel 2007 jest plik skoroszytu w formacie *XLTX*, ale Excel potrafi również otwierać i zapisywać pliki w wielu innych formatach. W tym podrozdziale dokonam szybkiego przeglądu typów plików, które są obsługiwane przez program Excel 2007.



Excel 2007 nie obsługuje już plików zapisanych w formatach arkuszy kalkulacyjnych Lotus lub Quattro.

Formaty plików obsługiwane w programie Excel

Excel 2007 domyślnie zapisuje pliki skoroszytów w zupełnie nowym formacie, ale nadal może otwierać i zapisywać pliki skoroszytów w formatach używanych przez poprzednie wersje Excela.



Aby zmienić domyślny typ zapisywanych plików, z menu *Office* wybierz polecenie *Opcje programu Excel*. Na ekranie pojawi się okno dialogowe *Opcje programu Excel*. Kliknij kategorię *Zapisywanie* i z listy rozwijanej *Zapisz pliki w następującym formacie* wybierz nowy, domyślny format zapisywania plików.

W tabeli 4.2 przedstawiono zestawienie typów plików obsługiwanych w programie Excel 2007. Pamiętaj, że zarówno pliki skoroszytów, jak i dodatków Excela mogą mieć dowolne rozszerzenia — innymi słowy, do zapisywania takich plików nie musisz używać rozszerzeń przedstawionych w poniższej tabeli.



Użytkownicy pakietów Microsoft Office XP oraz Microsoft Office 2003 mogą zainstalować *Pakiet zgodności formatu plików pakietu Microsoft Office 2007*, który pozwala na otwieranie i zapisywanie dokumentów w formatach używanych przez pakiet Office 2007. *Pakiet zgodności formatu plików pakietu Microsoft Office 2007* możesz pobrać ze strony <http://office.microsoft.com>.

Formaty plików tekstowych

Gdy spróbujesz otworzyć w Excelu plik tekstowy, na ekranie może pojawić się okno *Kreatora importu tekstu*, który pomoże w określeniu sposobu wczytania pliku.



Aby pominąć *Kreatora importu tekstu*, należy przed kliknięciem *OK* w oknie dialogowym *Otwieranie* nacisnąć klawisz *Shift*.

W tabeli 4.3. przedstawiono zestawienie typów plików tekstowych obsługiwanych w programie Excel 2007. Pliki tekstowe mogą przechowywać tylko dane z jednego arkusza.

Formaty plików baz danych

W tabeli 4.4 zamieszczono zestawienie typów plików baz danych obsługiwanych przez Excela 2007. Pliki baz danych mogą przechowywać tylko dane z jednego arkusza.

Tabela 4.2. Typy plików obsługiwanych w programie Excel 2007

Typ pliku	Rozszerzenie	Odczytywanie/ zapisywanie	Uwagi
<i>Skoroszyt programu Excel</i>	<i>xlsx</i>	Tak/Tak	Domyślny format plików skoroszytów Excela 2007. Nie może przechowywać makr VBA ani XLM.
<i>Skoroszyt programu Excel z obsługą makr</i>	<i>xlsm</i>	Tak/Tak	Domyślny format plików skoroszytów Excela 2007, w którym można przechowywać makra VBA i XLM.
<i>Skoroszyt binarny programu Excel</i>	<i>xlsb</i>	Tak/Tak	Binarny format plików skoroszytów Excela 2007 (BIFF12). Jest to zaktualizowana wersja poprzedniego formatu XLS (BIFF8).
<i>Szablon programu Excel</i>	<i>xltx</i>	Tak/Tak	Domyślny format szablonów Excela 2007. Nie może przechowywać makr VBA ani XLM.
<i>Szablon programu Excel z obsługą makr</i>	<i>xltxm</i>	Tak/Tak	Domyślny format szablonów Excela 2007, w którym można przechowywać makra.
<i>Dodatek programu Excel</i>	<i>xlam</i>	Tak/Tak	Domyślny format dodatków Excela 2007, w którym można przechowywać makra VBA i XLM.
<i>Skoroszyt programu Excel 97-2003</i>	<i>xls</i>	Tak/Tak	Binarny format plików skoroszytów (BIFF8), kompatybilny z plikami Excela w wersji od 97 do 2003.
<i>Szablon programu Excel 97-2003</i>	<i>xlt</i>	Tak/Tak	Binarny format plików szablonów (BIFF8), kompatybilny z plikami Excela w wersji od 97 do 2003.
<i>Dodatek programu Excel 97-2003</i>	<i>xla</i>	Tak/Tak	Binarny format plików dodatków (BIFF8), kompatybilny z plikami Excela w wersji od 97 do 2003.
<i>Skoroszyt Microsoft Excel 5.0/95</i>	<i>xls</i>	Tak/Tak	Binarny format plików skoroszytów (BIFF5), kompatybilny z plikami Excela w wersjach 5.0 i 95.
<i>Arkusz kalkulacyjny XML 2003</i>	<i>xml</i>	Tak/Tak	Format skoroszytów Microsoft XML 2003 (XMLSS).
<i>Dane XML</i>	<i>xml</i>	Tak/Tak	Ogólny format plików XML z danymi.

Inne formaty plików

W tabeli 4.5 zamieszczono zestawienie innych formatów plików baz danych obsługiwanych przez Excela 2007.

Tabela 4.3. Formaty plików tekstowych obsługiwanych przez Excela 2007

Typ pliku	Rozszerzenie	Odczytywanie/ zapisywanie	Uwagi
CSV (rozdzielany przecinkami)	csv	Tak/Tak	Kolumny danych są rozdzielane przecinkami, a wiersze znakami powrotu karetki. Excel może eksportować pliki CSV w subformacie Macintosh lub MS-DOS.
Tekst z formatowaniem	prn	Tak/Tak	Kolumny danych są rozdzielane znakami spacji, a wiersze znakami powrotu karetki.
Tekst	txt	Tak/Tak	Kolumny danych są rozdzielane znakami tabulacji, a wiersze znakami powrotu karetki. Excel może eksportować pliki tekstowe w subformacie Macintosh lub MS-DOS.
DIF (Format wymiany danych)	dif	Tak/Tak	Format plików używany przez program VisiCalc.
SYLK (Łącze symboliczne)	slk	Tak/Tak	Format plików używany przez program Multiplan.

Tabela 4.4. Formaty plików baz danych obsługiwanych przez Excela 2007

Typ pliku	Rozszerzenie	Odczytywanie/ zapisywanie	Uwagi
Access	mdb, mde, accdb, accde	Tak/Nie	W Excelu możesz otworzyć wybraną tabelę bazy danych.
DBASE	dbf	Tak/Nie	Format plików opracowany przez firmę Ashton-Tate, twórcę bazy danych dBASE.
Inne	Różne	Tak/Nie	Poprzez użycie poleceń znajdujących się w grupie <i>Dane zewnętrzne</i> na karcie <i>Dane</i> możesz importować dane z różnych źródeł zewnętrznych.

Pliki obszaru roboczego

Plik obszaru roboczego jest specjalnym plikiem zawierającym informacje na temat obszaru roboczego Excela. Jeżeli na przykład stworzyłeś projekt korzystający z dwóch skoroszytów i chciałbyś mieć ustawione ich okna w określony sposób, możesz zapisać ich układ w specjalnym pliku XLW. Później po każdym otwarciu takiego pliku XLW Excel przywróci żądany obszar roboczy.

Aby zapisać plik obszaru roboczego, przejdź na kartę *Widok*, naciśnij przycisk *Zapisz obszar roboczy*, znajdujący się w grupie opcji *Okno*, i kiedy Excel o to poprosi, podaj nazwę zapisywanego pliku.

Aby otworzyć plik obszaru roboczego, z menu *Office* wybierz polecenie i w oknie dialogowym *Otwieranie* z listy rozwijanej *Pliki typu* wybierz opcję *Obszary robocze (*.xlw)*.

Pamiętaj, że pliki obszaru roboczego *nie* zawierają żadnych skoroszytów, a jedynie informacje konfiguracyjne o układzie i widoczności okien skoroszytów. A zatem jeżeli zamierzasz przesłać konfigurację obszaru roboczego komuś innemu, nie zapomnij o wysłaniu wraz z plikiem XLW odpowiednich skoroszytów.

Tabela 4.5. Inne typy plików obsługiwane przez Excela 2007

Typ pliku	Rozszerzenie	Odczytywanie/ zapisywanie	Uwagi
<i>Plik HTML (Hypertext Markup Language)</i>	<i>htm, html</i>	Tak/Tak	Excel 2007 już nie obsługuje „podróży w obie strony” dla dokumentów zapisywanych w formacie HTML ¹ .
<i>Archiwum sieci Web, osobny plik</i>	<i>mht, mhtml</i>	Tak/Tak	Pliki zapisane w tym formacie mogą być odczytywane tylko w przeglądarce Microsoft Internet Explorer.
<i>Plik PDF (Portable Document Format)</i>	<i>pdf</i>	Nie/Tak	Format pliku opracowany przez firmę Adobe. Wymaga zainstalowania w Excelu odpowiedniego, bezpłatnego dodatku, który można pobrać ze strony internetowej firmy Microsoft.
<i>Plik XPS (XML Paper Specification)</i>	<i>xml</i>	Nie/Tak	Alternatywa firmy Microsoft dla plików PDF firmy Adobe. Wymaga zainstalowania w Excelu odpowiedniego, bezpłatnego dodatku, który można pobrać ze strony internetowej firmy Microsoft.

Praca z plikami szablonów

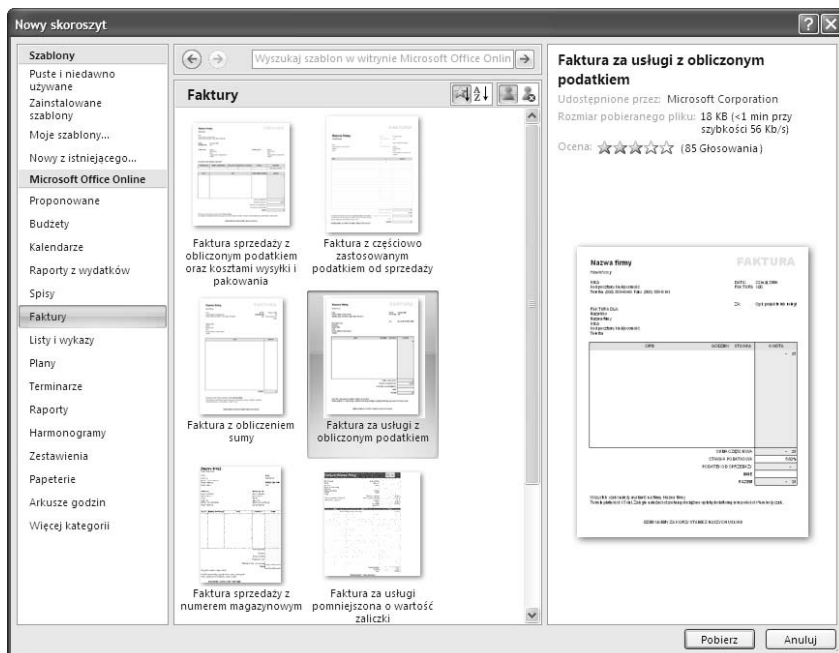
Szablon to, jak sama nazwa wskazuje, model, na bazie którego tworzymy coś innego. Szablon programu Excel to skoroszyt, którego używasz do tworzenia innych skoroszytów. Jako plik szablonu (z rozszerzeniem *XLTX*) może zostać zapisany dowolny skoroszyt. Szablony są szczególnie przydatne w przypadku częstego tworzenia podobnych plików. Aby na przykład co miesiąc generować podobne raporty sprzedaży i przy okazji zaoszczędzić czas, należy utworzyć szablon przechowujący formuły i wykresy wymagane do wygenerowania raportu. Po stworzeniu plików w oparciu o nowy szablon wystarczy jedynie wprowadzić odpowiednie wartości.

Przeglądanie dostępnych szablonów

Excel 2007 daje Ci do dyspozycji bardzo wiele różnych szablonów. Aby przeglądać galerię szablonów, z menu *Office* wybierz polecenie *Nowy*. Na ekranie pojawi się okno dialogowe *Nowy skoroszyt*. Na panelu w lewej części okna znajdziesz różne kategorie szablonów, a w prawej części okna pojawi się lista szablonów, z których niedawno korzystałeś.

W sekcji *Microsoft Office Online* znajdziesz wiele różnych kategorii szablonów. Po kliknięciu wybranej kategorii w prawej części okna pojawi się lista dostępnych szablonów. Aby użyć wybranego szablonu, zaznacz go i naciśnij przycisk *Pobierz*. Na rysunku 4.3 przedstawiono niektóre szablony dostępne w kategorii *Faktury*.

¹ Oznacza to, że nie można zapisać dokumentu w formacie HTML i następnie odczytać go w Excelu bez utraty funkcjonalności skoroszytu — *przyp. tłum.*



Rysunek 4.3. Szablony, których możesz użyć do tworzenia faktur

Na witrynie internetowej *Microsoft Office Online* znajdziesz bardzo wiele różnych szablonów, niektóre z nich będą znacznie lepsze niż inne. Nie przejmuj się, jeżeli zdarzy Ci się kilka razy natrafić na niezbyt udane szablony. Nawet takie szablony czasami mogą być użyteczne po kilku drobnych modyfikacjach. Modyfikacja istniejącego szablonu jest z reguły znacznie łatwiejsza niż tworzenie nowego skoroszytu od podstaw.



Lokalizacja katalogu *Szablony* zmienia się w zależności od wersji Excela. Aby znaleźć położenie tego foldera, należy wykonać następującą instrukcję języka VBA:

```
MsgBox Application.TemplatesPath
```

Tworzenie szablonów

Excel obsługuje trzy rodzaje szablonów:

- *Domyślny szablon skoroszytu* — jest wykorzystywany jako baza dla nowych skoroszytów. Plik domyślnego szablonu skoroszytu nosi nazwę *book.xltx*.
- *Domyślny szablon arkusza* — jest wykorzystywany jako baza dla nowych arkuszy wstawianych do skoroszytu. Plik domyślnego szablonu arkusza nosi nazwę *sheet.xltx*.
- *Własne szablony skoroszytów* — zazwyczaj są to gotowe do użycia skoroszyty, zawierające odpowiednie formuły, ale w praktyce szablony mogą być zarówno bardzo proste, jak i bardzo złożone. Zazwyczaj szablony są tworzone w taki sposób, że po ich otwarciu użytkownik musi tylko wprowadzić odpowiednie dane i od razu otrzymuje żądany rezultat.

Tworzenie szablonu zmieniającego domyślne ustawienia skoroszytów

Każdy nowy skoroszyt posiada pewne predefiniowane ustawienia domyślne. Na przykład nowy skoroszyt posiada trzy arkusze, linie siatki są wyświetlane, domyślna czcionka to Calibri o rozmiarze 11 punktów, domyślna szerokość kolumny to 8,43 punktu i tak dalej. Jeżeli któreś z tych ustawień Ci nie odpowiada, możesz je w prosty sposób zmienić.

Wprowadzanie zmian domyślnych ustawień programu Excel jest zadaniem stosunkowo prostym i w perspektywie długiego korzystania z programu Excel może Ci zaoszczędzić sporo czasu. Aby zmienić ustawienia domyślne, powinieneś wykonać polecenia opisane poniżej:

1. Otwórz nowy skoroszyt.
2. Dodaj lub usuń odpowiednią liczbę arkuszy, tak aby osiągnąć ich żadaną ilość.
3. Dokonaj odpowiednich zmian i modyfikacji ustawień skoroszytu, takich jak szerokość kolumny, style formatowania, opcje ustawienia strony i wiele innych ustawień dostępnych w oknie *Opcje programu Excel*.

Aby zmienić domyślny sposób formatowania komórek, przejdź na kartę *Narzędzia główne* i naciśnij przycisk *Style komórki*, znajdujący się w grupie opcji *Style*, a następnie zmień ustawienia stylu *Normalny* (na przykład możesz zmienić domyślny krój czcionki, rozmiar czcionki czy sposób formatowania liczb).

4. Po zakończeniu wprowadzania modyfikacji wybierz z menu *Office* polecenie *Zapisz jako*.
5. Na ekranie pojawi się okno dialogowe *Zapisywanie jako*. Z listy rozwijanej *Zapisz jako typ* wybierz opcję *Szablon programu Excel (*.xltx)*.
6. W polu *Nazwa pliku* wpisz nazwę *book.xltx*.
7. Zapisz plik w folderze *\XLStart* (a nie w domyślnym folderze *Szablony*).
8. Zamknij plik.



Folder *\XLStart* może się znajdować w następujących lokalizacjach:

C:\Documents and Settings\<nazwa_użytkownika>\Dane aplikacji\Microsoft\Excel\XLStart
C:\Program Files\Microsoft Office\Office12\XLStart

Aby określić rzeczywistą lokalizację tego foldera, powinieneś wykonać następujące polecenie VBA:

```
MsgBox Application.StartupPath
```

Po wykonaniu poleceń opisanych powyżej nowy skoroszyt, który będzie się pojawiał po uruchomieniu Excela, będzie oparty na ustawieniach szablonu *book.xltx*. Aby utworzyć nowy skoroszyt oparty na tym szablonie, możesz po prostu nacisnąć kombinację klawiszy *Ctrl+N*. Jeżeli będziesz chciał kiedyś powrócić do „fabrycznych”, domyślnych ustawień skoroszytu Excela, po prostu usuń plik *book.xltx*.



Jeżeli z menu *Office* wybierzesz polecenie *Nowy* i w oknie dialogowym *Nowy skoroszyt* wybierzesz opcję *Pusty skoroszyt*, utworzony skoroszyt nie będzie oparty na szablonie *book.xltx*. Szczepnie mówiąc, nie wiem, czy wynika to z błędu Excela, czy jest po prostu działaniem zgodnym z intencjami programistów firmy Microsoft.

Tworzenie szablonu zmieniającego domyślne ustawienia arkusza

Kiedy wstawiasz nowy arkusz do skoroszytu, Excel tworzy go na bazie predefiniowanych ustawień domyślnych, obejmujących takie elementy jak szerokość kolumn, wysokość wierszy i tak dalej. Jeżeli któreś z tych ustawień domyślnych Ci nie odpowiada, możesz je w prosty sposób zmienić. Aby to zrobić, powinieneś wykonać polecenia opisane poniżej:

1. Utwórz nowy skoroszyt i usuń z niego wszystkie arkusze z wyjątkiem jednego.
2. Dokonaj odpowiednich zmian i modyfikacji ustawień skoroszytu, takich jak szerokość kolumny, style formatowania, opcje ustawienia strony i wiele innych ustawień dostępnych w oknie *Opcje programu Excel*.
3. Po zakończeniu wprowadzania modyfikacji wybierz z menu *Office* polecenie *Zapisz jako*.
4. Na ekranie pojawi się okno dialogowe *Zapisywanie jako*. Z listy rozwijanej *Zapisz jako typ* wybierz opcję *Szablon programu Excel (*.xltx)*.
5. W polu *Nazwa pliku* wpisz nazwę *sheet.xltx*.
6. Zapisz plik w folderze *\XLStart* (a nie w domyślnym folderze *Szablony*).
7. Zamknij plik.
8. Zamknij i ponownie uruchom Excela.

Po wykonaniu poleceń opisanych powyżej wszystkie nowe arkusze wstawiane do skoroszytu za pomocą polecenia *Wstaw arkusz* (przycisk tego polecenia znajdziesz po prawej stronie ostatniej karty arkusza) będą sformatowane w oparciu o szablon *sheet.xltx*. Aby wstawić nowy arkusz, możesz również nacisnąć kombinację klawiszy *Shift+F11*.

Tworzenie własnych szablonów skoroszytu

Szablony *book.xltx* oraz *sheet.xltx*, o których była mowa w poprzedniej sekcji, to dwa specjalne rodzaje szablonów odpowiedzialne za domyślne ustawienia odpowiednio nowych skoroszytów i nowych arkuszy. W tym podrozdziale omówię rodzaje szablonów skoroszytów, określanych po prostu jako szablony skoroszytów użytkownika, które są tworzone jako baza dla dokumentów, z jakimi pracujesz na co dzień.

Po co używać szablonów skoroszytów? Odpowiedź jest prosta — po to, aby zaoszczędzić sobie konieczności ciągłego wykonywania tej samej pracy. Załóżmy, że stworzysz comiesięczny raport sprzedaży, składający się z informacji o poziomie sprzedaży w poszczególnych regionach oraz pewnej liczby obliczeń i wykresów. W takiej sytuacji możesz utworzyć szablon zawierający wszystkie wspomniane elementy z wyjątkiem konkretnych wartości. Teraz, kiedy nadejdzie czas utworzenia raportu, możesz po prostu utworzyć nowy skoroszyt oparty na szablonie raportu, wprowadzić aktualne dane i raport będzie gotowy.



Oczywiście w podobnym celu możesz wykorzystać po prostu raport z poprzedniego miesiąca i tylko zmienić odpowiednie dane, ale taki sposób postępowania jest bardzo podatny na powstawanie błędów, a co gorsza, jeżeli zapomnisz użyć polecenia *Zapisz jako* i zapisać nowy raport pod inną nazwą, możesz w niezamierzony sposób nadpisać raport z poprzedniego miesiąca i utracić tamte dane. Innym rozwiązaniem jest użycie polecenia *Nowy z istniejącego*, które znajdziesz w oknie dialogowym *Nowy skoroszyt*. Wykonanie tego polecenia powoduje utworzenie nowego skoroszytu na bazie skoroszytu istniejącego i nadaje mu inną nazwę, dzięki czemu oryginalny skoroszyt pozostanie nienaruszony.

Kiedy tworzysz skoroszyt w oparciu o szablon, domyślna nazwa skoroszytu składa się z nazwy szablonu i kolejnego numeru skoroszytu. Jeśli na przykład tworzysz nowy skoroszyt oparty na szablonie *Raport sprzedaży.xltx*, to domyślną nazwą skoroszytu będzie *Raport sprzedaży1.xlsx*. Kiedy po raz pierwszy będziesz chciał zapisać skoroszyt utworzony na bazie szablonu, Excel wyświetli na ekranie okno dialogowe *Zapisywanie jako* i będziesz mógł nadać skoroszytowi nową nazwę.

Szablon skoroszytu użytkownika jest w zasadzie normalnym skoroszytem Excela, stąd możesz w nim używać wszystkich dostępnych elementów, takich jak na przykład wykresy, formuły czy makra. Zazwyczaj szablony są tworzone w taki sposób, że po ich otwarciu użytkownik musi tylko wprowadzić odpowiednie dane i od razu otrzymuje żądany rezultat — innymi słowy, większość szablonów zawiera wszystko z wyjątkiem danych, które musi wprowadzić użytkownik.



Jeżeli szablon zawiera makra, musi zostać zapisany w formacie *Szablon programu Excel z obsługą makr*. Takie pliki szablonów mają rozszerzenie **.xltn*.

Budowa plików programu Excel

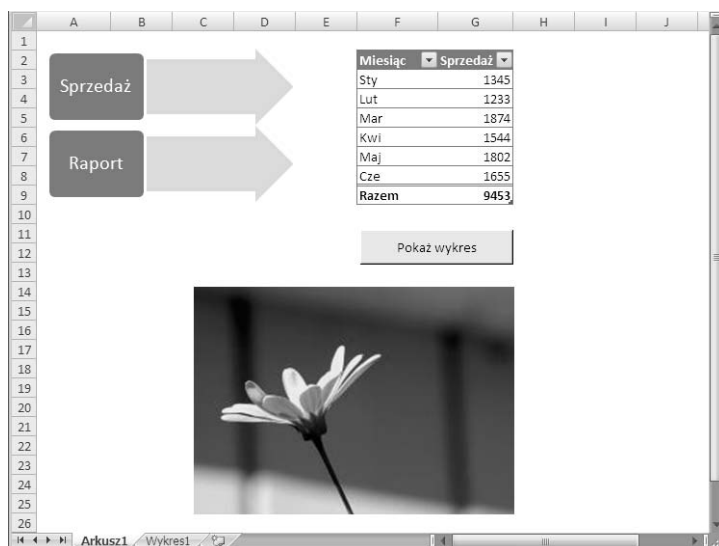
Jak już wspominaliśmy, Excel 2007 do zapisywania swoich skoroszytów, szablonów i dodatków używa zupełnie nowego formatu plików XML. Tak naprawdę są to pliki w formacie archiwum ZIP, a zatem ich zawartość może zostać „wypakowana” i sprawdzona niejako „na zewnątrz”.

Poprzednie wersje programu Excel do zapisu dokumentów używały plików binarnych. Pomimo iż specyfikacja tych plików jest znana, to jednak praca z nimi nie należała do najłatwiejszych. Nowy format dokumentów Excela 2007, XML, jest formatem „otwartym”, co oznacza, że takie pliki mogą być również otwierane i przetwarzane przy użyciu innego oprogramowania.

Zagłębiamy do wnętrza pliku

W tym podrozdziale omówimy elementy składowe, które możesz znaleźć wewnątrz typowego pliku *XLSM* programu Excel (czyli pliku skoroszytu z obsługą makr). Wygląd naszego przykładowego skoroszytu, o nazwie *przyklad.xlsm*, został przedstawiony na rysunku 4.4. Skoroszyt składa się z jednego arkusza danych, jednego arkusza wykresu oraz prostego makra VBA. Na arkuszu znajduje się tabela, przycisk (formant formularza), diagram składający się z obiektów *SmartArt* oraz zdjęcie przedstawiające kwiat.

Rysunek 4.4.
Przykład prostego
skoroszytu



Skoroszyt z tym przykładem (*przykład.xlsm*) znajdziesz na płycie CD-ROM dołączonej do książki.

Aby zajrzeć „pod maskę” pliku skoroszytu programu Excel 2007, będziesz musiał uruchomić program Windows Explorer, odszukać nasz plik skoroszytu i dodać mu rozszerzenie ZIP. Po wykonaniu tej operacji zamiast pliku *przykład.xlsm* powinieneś mieć plik *przykład.xlsm.zip*. Teraz możesz otworzyć ten plik w dowolnym programie obsługującym archiwa ZIP, ale równie dobrze możesz po prostu skorzystać z obsługi plików ZIP wbudowanej w systemie Windows XP.



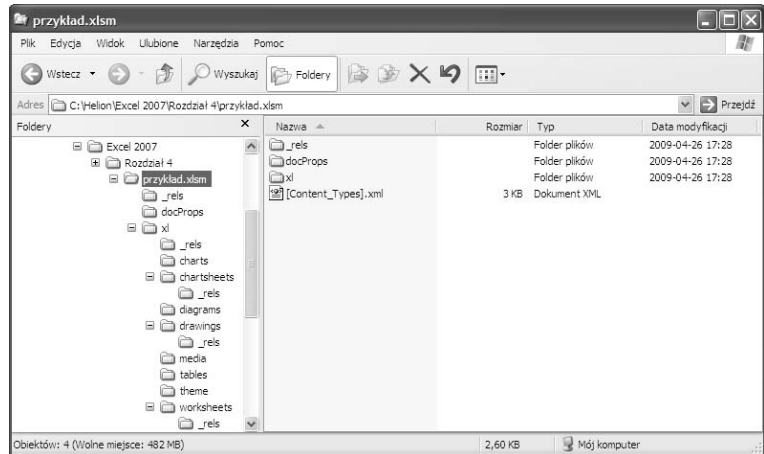
Lepszym rozwiązaniem może być wypakowanie zawartości archiwum do osobnego, nieskompresowanego foldera na dysku, co powinno zdecydowanie ułatwić przeglądanie poszczególnych plików składowych. Aby to zrobić, kliknij plik skoroszytu prawym przyciskiem myszy i z menu podręcznego wybierz polecenie *Wyodrębnij wszystkie*.

Pierwszą rzeczą, która rzuca się w oczy, jest to, że plik posiada strukturę katalogu. Na rysunku 4.5 w lewym panelu widać pełną strukturę wewnętrzną pliku skoroszytu. Oczywiście dokładna struktura pliku i jego zawartość będzie się różniła w zależności od skoroszytu.

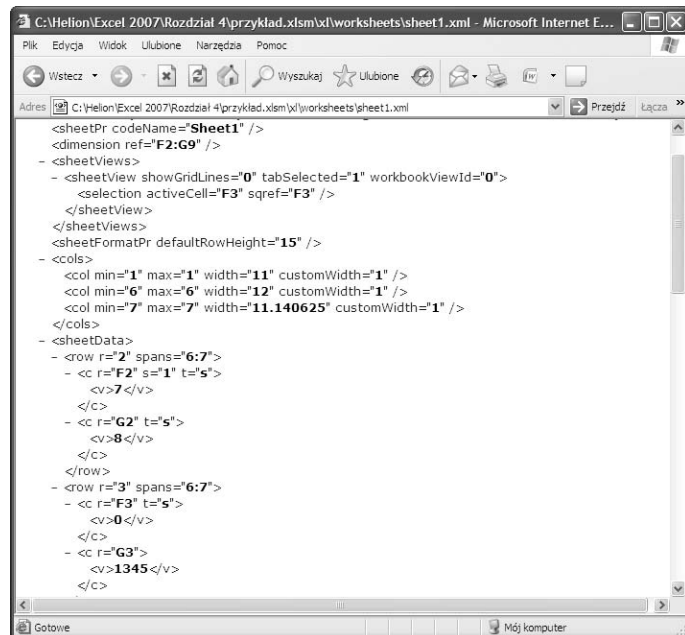
Oprócz kilku wyjątków zdecydowana większość plików w skoroszytcie to pliki tekstowe, a dokładniej mówiąc, pliki w formacie XML. Ich zawartość możesz przeglądać w dowolnym edytorze tekstu, edytorze XML, przeglądarce sieciowej, a nawet w Excelu. Na rysunku 4.6 przedstawiono zawartość jednego z plików, wyświetloną w oknie przeglądarki Internet Explorer. Pliki w formatach innych niż XML są powiązane z obiektami graficznymi i projektami VBA (które są zapisywane w formacie binarnym).

Ten plik XML posiada trzy foldery główne i kilka subfolderów. Zwróć uwagę, że wiele folderów w skoroszytcie posiada podkatalog o nazwie *_rels*. W tym podkatalogu znajduje się plik XML, który definiuje powiązania tego elementu z innymi elementami składowymi pakietu.

Rysunek 4.5.
Wewnętrzna struktura
katalogów pliku
skoroszytu



Rysunek 4.6.
Przeglądanie
zawartości pliku XML
w przeglądarce
sieciowej



Poniżej zamieszczono listę folderów, które możesz znaleźć w naszym skoroszytce *przyklad.xlsm*:

- **_rels** — zawiera informacje o relacjach pomiędzy elementami składowymi skoroszytu.
- **docProps** — zawiera plik XML opisujący właściwości plików i ustawienia aplikacji.
- **xl** — w tym folderze przechowywane są główne elementy składowe skoroszytu. Nazwa tego foldera zmienia się w zależności od aplikacji (*xl*, *ppt*, *word* i tak dalej). Znajdziesz tutaj kilka plików XML zawierających ustawienia skoroszytu, a jeżeli

skoroszyt zawiera kod VBA, w tym folderze znajdzie się również odpowiedni plik binarny z rozszerzeniem *BIN*. W folderze *xl* znajdziesz kilka podkatalogów (ich nazwy i liczba mogą się zmieniać w zależności od zawartości skoroszytu):

- *charts* — zawiera pliki *XML* opisujące wykresy znajdujące się w skoroszycie (po jednym pliku dla każdego wykresu). W plikach *XML* znajdują się ustawienia poszczególnych wykresów.
- *chartsheets* — zawiera pliki *XML* z danymi dla wykresów umieszczonych na poszczególnych arkuszach wykresów.
- *diagrams* — zawiera pliki *XML* opisujące diagramy (utworzone przy użyciu obiektów *SmartArt*) w skoroszycie.
- *drawings* — zawiera pliki *XML* opisujące poszczególne „rysunki”, takie jak przyciski, wykresy czy obrazy.
- *media* — zawiera osadzone pliki multimedialne, takie jak na przykład pliki *GIF* czy *JPEG*.
- *tables* — zawiera pliki *XML* opisujące poszczególne tabele danych.
- *theme* — zawiera plik *XML* z danymi o motywie skoroszytu.
- *worksheets* — zawiera pliki *XML* opisujące poszczególne arkusze skoroszytu.



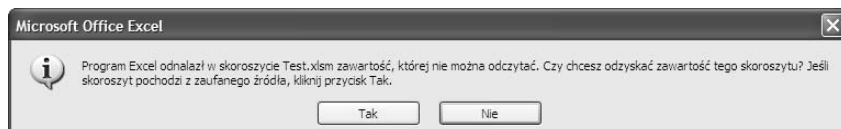
Wskazówka

Jeżeli dodasz do pliku skoroszytu rozszerzenie *ZIP*, to i tak nadal będziesz mógł otworzyć taki plik w programie Excel. Excel nie zwraca specjalnej uwagi na rozszerzenia plików. W razie potrzeby możesz od razu zapisać skoroszyt na dysku w postaci pliku z rozszerzeniem *ZIP*. Aby to zrobić, po prostu dodaj w oknie dialogowym *Zapisywanie* jako do nazwy pliku rozszerzenie *ZIP* i następnie umieść całą nazwę w cudzysłowie, na przykład *"MójSkoroszyt.xlsx.zip"*.

Dlaczego format pliku jest taki ważny?

Nowe, „otwarte” formaty *XML* plików wykorzystywanych przez aplikacje Microsoft Office reprezentują znaczący postęp dla całej społeczności użytkowników komputerów. Po raz pierwszy w historii można względnie łatwo odczytywać i zapisywać skoroszyty Excela przy użyciu oprogramowania innego niż sam Excel. Na przykład możesz napisać program, który będzie modyfikował zawartość tysięcy skoroszytów Excela, bez potrzeby uruchamiania samego programu Excel. Taki program mógłby na przykład wstawiać do każdego ze skoroszytów nowy arkusz. Programista tworzący taką aplikację musiałby oczywiście wykazać się dogłębną znajomością struktur *XML* plików skoroszytów, ale takie zadanie nie jest przecież takie trudne.

Co ważne, nowe formaty plików są znacznie mniej podatne na uszkodzenia (w porównaniu ze starymi formatami binarnymi). Aby to przetestować, zapisałem skoroszyt i następnie usunąłem z niego jeden z plików *XML* opisujących arkusze. Kiedy spróbowałem otworzyć taki uszkodzony plik w Excelu, na ekranie pojawiło się okno z komunikatem, przedstawione na rysunku 4.7. Dzięki analizie danych zapisanych w plikach *.res* Excel był w stanie stwierdzić, że plik został uszkodzony. W tym przypadku Excel był w stanie naprawić zawartość pliku i poprawnie go otworzyć. Usunięty arkusz został wstawiony do skoroszytu, ale zniknęły znajdujące się na nim dane.



Rysunek 4.7. Excel bardzo często jest w stanie samodzielnie naprawić uszkodzone pliki

Kolejną zaletą nowego formatu plików Excela jest to, że spakowane pliki XML są zazwyczaj znacznie mniejsze niż odpowiadające im pliki binarne w starym formacie. Dodatkowo strukturalna budowa plików skoroszytu pozwala na wyodrębnianie z nich poszczególnych elementów (na przykład wszystkich elementów graficznych).

Typowy użytkownik Excela prawdopodobnie nigdy nie będzie musiał przeglądać ani modyfikować elementów składowych XML pliku skoroszytu. Programiści tworzący aplikacje Excela mogą jednak wykorzystywać tę wiedzę do utworzenia kodu modyfikującego Wstążkę, czyli graficzny interfejs użytkownika programu Excel. Aby można było to zrobić, dogłębna znajomość struktur XML pliku skoroszytu jest po prostu nieodzowna.



Więcej szczegółowych informacji na temat modyfikacji Wstążki Excela znajdziesz w rozdziale 22.

Plik QAT

Jedynym elementem nowego interfejsu programu Excel 2007, który może być w prosty sposób modyfikowany przez użytkownika, jest pasek narzędzi *Szybki dostęp*. Informacje dotyczące konfiguracji tego paska są zapisane w pliku *excel.qat*, zlokalizowanym w następującym katalogu:

```
C:\Documents and Settings\<nazwa_użytkownika>\Ustawienia lokalne\
Dane aplikacji\Microsoft\OFFICE
```

Ten plik jest aktualizowany za każdym razem, kiedy użytkownik zmienia konfigurację paska narzędzi *Szybki dostęp*. Zmiany w pliku są dokonywane natychmiast, a nie tylko w momencie, kiedy Excel jest zamykany.

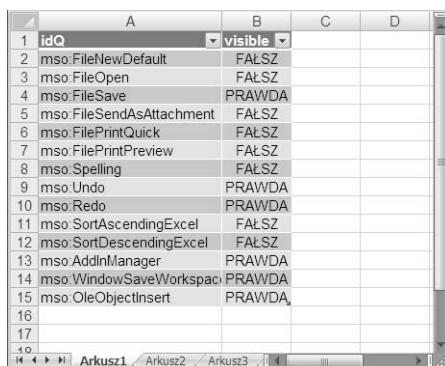
Excel.qat jest plikiem w formacie XML, a zatem możesz przeglądać jego zawartość przy użyciu edytora XML, przeglądarki sieciowej, a nawet Excela. Aby wyświetlić zawartość tego pliku w Excelu, powinieneś wykonać następujące polecenia:

1. Zrób kopię pliku *excel.qat*.
2. Dodaj do nazwy kopii pliku rozszerzenie XML, tak aby jego nazwa wyglądała na przykład tak: *KopiaExcel.qat.xml*.
3. Otwórz plik za pomocą polecenia *Office/Otwórz*. Zamiast tego możesz po prostu przeciągnąć plik do okna Excela.
4. Na ekranie pojawi się okno dialogowe *Otwieranie pliku XML*, zawierające kilka opcji; wybierz opcję *Jako tabelę XML* i naciśnij przycisk *OK*.

Na rysunku 4.8 przedstawiono zawartość pliku *Excel.qat*, który został zaimportowany do Excela i wyświetlony w formie tabeli. Jak widać, oprócz jedenastu domyślnych poleceń na pasku znajdują się trzy dodatkowe polecenia. Spośród poleceń domyślnych widoczne są tylko trzy.

Rysunek 4.8.

Przeglądanie w Excelu zawartości pliku konfiguracyjnego paska narzędzi Szybki dostęp



	A	B	C	D
1	idQ	visible		
2	mso.FileNewDefault	FALSZ		
3	mso.FileOpen	FALSZ		
4	mso.FileSave	PRAWDA		
5	mso.FileSendAsAttachment	FALSZ		
6	mso.FilePrintQuick	FALSZ		
7	mso.FilePrintPreview	FALSZ		
8	mso.Spelling	FALSZ		
9	mso.Undo	PRAWDA		
10	mso.Redo	PRAWDA		
11	mso.SortAscendingExcel	FALSZ		
12	mso.SortDescendingExcel	FALSZ		
13	mso.AddInManager	PRAWDA		
14	mso.WindowSaveWorkspac	PRAWDA		
15	mso.OleObjectInsert	PRAWDA		
16				
17				

Konfigurację paska narzędzi *Szybki dostęp* możesz udostępniać innym użytkownikom. Jeśli na przykład upakowałeś na pasku narzędzi *Szybki dostęp* ze dwa tuziny użytecznych poleceń i zrobiło to ogromne wrażenie na Twoich kolegach z pracy, możesz przekazać im kopię swojego pliku *Excel.qat* i powiedzieć, gdzie mają go umieścić.

Pamiętaj, nigdy nie próbuj samodzielnie modyfikować zawartości pliku *Excel.qat*, jeśli nie jesteś całkowicie przekonany, że wiesz, co robisz.

Plik XLB

Konfigurację pasków narzędzi i menu Excel zapisuje w pliku formatu *XLB*. Pomimo iż Excel 2007 oficjalnie nie obsługuje niestandardowych pasków narzędzi i menu użytkownika w sposób, jaki miał miejsce w poprzednich wersjach Excela, to jednak nadal korzysta z pliku *XLB*.

Po zamknięciu Excela aktualna konfiguracja pasków narzędzi jest przechowywana w pliku *Excel12.xlb*. Plik ten (najczęściej) jest przechowywany w następującej lokalizacji:

C:\Documents and Settings\<nazwa_użytkownika>\Dane aplikacji\Microsoft\Excel

Ten plik binarny zawiera informacje o położeniu i widoczności wszystkich niestandardowych pasków narzędzi i menu wraz z opisem modyfikacji wbudowanych pasków narzędzi i menu.

Pliki dodatków

Dodatek jest plikiem skoszytu różniącym się od standardowego skoszytu następującymi — ważnymi — elementami:

- Właściwość `IsAddin` skoroszytu ma wartość `True` — co oznacza, że można go załadować przy użyciu okna dialogowego *Dodatki*. Aby przywołać to okno na ekran, wybierz z menu *Office* polecenie *Opcje programu Excel*. Na ekranie pojawi się okno dialogowe *Opcje programu Excel*. Kliknij kategorię *Dodatki*, w prawym panelu rozwiń listę *Zarządzaj*, wybierz z listy opcję *Dodatki programu Excel* i naciśnij przycisk *Przejdź*.
- Skoroszyt dodatku jest ukryty i nie może zostać wyświetlony przez użytkownika — w konsekwencji dodatek może nigdy nie być aktywnym skoroszytem.
- Z punktu widzenia kodu VBA skoroszyt dodatku nie jest elementem kolekcji `Workbooks`.

Dodatki wzbogacają Excel o nowe funkcje lub właściwości, z których można korzystać tak, jakby były wbudowane w program.

Własne dodatki możesz tworzyć w oparciu o pliki standardowych skoroszytów. W praktyce dodatki są preferowaną metodą dystrybucji niektórych aplikacji przeznaczonych dla Excela. Domyślnie pliki dodatków programu Excel 2007 są zapisywane z rozszerzeniem *XLAM*.



Poza dodatkami o rozszerzeniu *.xlam* Excel obsługuje dodatki *XLL* i *COM*. Tego typu dodatki są tworzone przy użyciu innego oprogramowania niż Excel. W tej książce omówiono jedynie dodatki typu *XLAM*.



Więcej szczegółowych informacji na temat dodatków znajdziesz w rozdziale 21.

Ustawienia Excela w rejestrze systemu Windows

W oknie dialogowym *Opcje programu Excel* znajduje się kilka tuzinów opcji konfiguracyjnych Excela. Excel przechowuje te ustawienia w rejestrach systemu Windows i odczytuje je stamtąd podczas uruchamiania programu. W tym podrozdziale przedstawiono kilka podstawowych informacji na temat rejestru systemu Windows i wyjaśniono, w jaki sposób Excel korzysta z niego do przechowywania własnych ustawień.

Rejestr systemu Windows

Rejestr systemu Windows to, ogólnie mówiąc, centralna, hierarchiczna baza danych wykorzystywana zarówno przez system operacyjny, jak i jego aplikacje. Rejestr po raz pierwszy pojawił się w systemie Windows 95, zastępując pliki *INI*, w których do tej pory system operacyjny i aplikacje przechowywały swoje ustawienia.



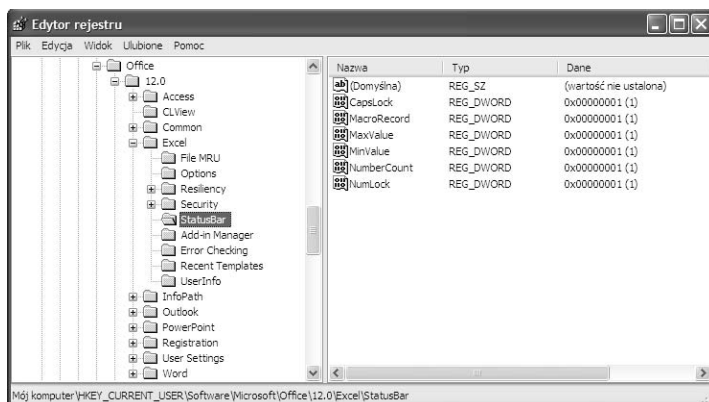
Makra języka VBA również mogą odczytywać i zapisywać informacje w rejestrze systemu Windows. Więcej szczegółowych informacji na ten temat znajdziesz w rozdziale 11.

Do przeglądania zawartości rejestru, a nawet modyfikacji jego zawartości (o ile wiesz, co robisz...) możesz użyć programu o nazwie *Edytor rejestru* (plik *regedit.exe*). Zanim

jednak rozpoczniesz eksplorację rejestru, powinieneś poświęcić chwilę na zapoznanie się z zawartością ramki „Zanim dokonasz zmian w rejestrze...”. Na rysunku 4.9 przedstawiono wygląd okna Edytora rejestru.

Rysunek 4.9.

Edytor rejestru pozwala na przeglądanie rejestru i wprowadzanie do niego zmian



Rejestr ma strukturę hierarchiczną i składa się z kluczy i wartości. Główne klucze rejestru systemu Windows są następujące:

- HKEY_CLASSES_ROOT,
- HKEY_CURRENT_USER,
- HKEY_LOCAL_MACHINE,
- HKEY_USERS,
- HKEY_CURRENT_CONFIG,
- HKEY_DYN_DATA.

Ustawienia Excela

Informacje używane przez Excela 2007 są przechowywane w następującej części rejestru:

HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Excel

Zanim dokonasz zmian w rejestrze...

Za pomocą programu *regedit.exe* możesz zmienić dowolną wartość w rejestrze systemu Windows, w tym informacje mające krytyczne znaczenie dla pracy i stabilności systemu operacyjnego. Krótko mówiąc, jeżeli zmienisz nie to, co trzeba, system Windows może po prostu przestać działać poprawnie.

Powinieneś wyrobić sobie nawyk wybierania z menu *Plik* programu Edytor rejestru polecenia *Eksportuj*. Polecenie to umożliwia zapisanie w formacie ASCII zawartości całego rejestru lub tylko wybranej części. Jeżeli stwierdzisz, że zrobiłeś coś nie tak, to w celu przywrócenia rejestru do poprzedniego stanu zawsze możesz zaimportować plik ASCII (z menu *Plik* należy wybrać pozycję *Importuj*). Więcej szczegółowych informacji na temat korzystania z edytora rejestru znajdziesz w jego systemie pomocy.

W tej sekcji rejestru znajduje się szereg kluczy zawierających specyficzne wartości wpływające na sposób działania Excela.

Ustawienia rejestru są automatycznie uaktualniane przez Excela w momencie jego zamykania.



Excel wczytuje zawartość rejestru systemu Windows tylko raz — podczas uruchamiania programu. Ustawienia rejestru uaktualniane są też tylko raz — gdy Excel zostanie prawidłowo zamknięty. Jeżeli Excel zawiesi się (co się niestety czasami zdarza), zawartość rejestru nie zostanie uaktualniona. Jeżeli na przykład zmodyfikujesz jedno z ustawień Excela, takie jak wyświetlanie paska statusu, dokonana zmiana nie zostanie zapisana w rejestrze do momentu poprawnego zamknięcia aplikacji.

W tabeli 4.6 przedstawiono sekcje rejestru, których używa Excel 2007. Pamiętaj, że na różnych komputerach zawartość tych sekcji rejestru może się różnić od siebie.

Tabela 4.6. Dane konfiguracyjne Excela przechowywane w rejestrze

Sekcja	Opis
<i>Add-in Manager</i>	Zawiera listę dodatków widocznych w oknie dialogowym <i>Dodatki</i> . Na liście nie są widoczne dodatki dołączone do Excela. Jeżeli na liście znajduje się wpis dotyczący dodatku, którego już nie używasz, możesz go usunąć za pomocą programu Edytor rejestru.
<i>Converters</i>	Zawiera listę dodatkowych (zewnętrznych) konwerterów plików, których nie wbudowano do Excela.
<i>Error Checking</i>	Przechowuje ustawienia dotyczące sprawdzania błędów w formułach.
<i>File MRU</i>	Przechowuje listę ostatnio używanych plików (która pojawia się w menu po naciśnięciu przycisku <i>Office</i>).
<i>Options</i>	Sekcja do przechowywania ustawień różnych.
<i>Recent Templates</i>	Przechowuje nazwy ostatnio używanych szablonów.
<i>Resiliency</i>	Informacje używane do przywracania dokumentów.
<i>Security</i>	Określa poziom bezpieczeństwa otwieranych plików zawierających makra.
<i>Spell Checker</i>	Przechowuje informacje o opcji modułu sprawdzającego pisownię.
<i>StatusBar</i>	Przechowuje ustawienia paska stanu.
<i>UserInfo</i>	Przechowuje informacje dotyczące użytkownika.

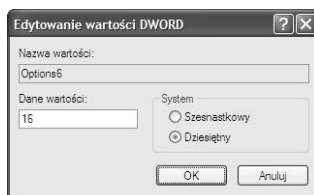
Co prawda większość ustawień Excela można zmienić za pośrednictwem okna dialogowego *Opcje programu Excel*, ale kilka przydatnych kluczy nie może zostać w ten sposób zmodyfikowanych (w tym przypadku można posłużyć się programem *Edytor rejestru*). Na przykład założymy, że kiedy zaznaczasz zakres komórek, chciałbyś, aby zaznaczone komórki były wyróżnione bardziej kontrastowo, czarno na białym. Takiej zmiany nie można dokonać z poziomu Excela, ale możesz to osiągnąć, dodając do rejestru odpowiedni klucz. Aby to zrobić, wykonaj polecenia opisane poniżej:

1. Uruchom *Edytor rejestru* i odszukaj następującą sekcję:

HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Excel\Options

2. Kliknij nazwę sekcji prawym przyciskiem myszy i z menu podręcznego wybierz polecenie *Nowy/Wartość DWORD*.
3. Zmień nazwę utworzonego elementu na *Options6*.
4. Kliknij element *Options6* prawym przyciskiem myszy i z menu podręcznego wybierz polecenie *Modyfikuj*.
5. Na ekranie pojawi się okno dialogowe *Edytowanie wartości DWORD*. Wybierz opcję *Dziesiętny* i w polu *Dane wartości* wpisz wartość 16 (patrz rysunek 4.10).

Rysunek 4.10.
*Ustawianie wartości
klucza rejestru*



Kiedy zamkniesz i ponownie uruchomisz Excela, zaznaczone obszary będą wyróżniane kontrastowym, czarnym tłem, zamiast dotychczasowej szarości. Jeżeli taki efekt Ci się nie spodoba, po prostu usuń wpis *Options6* z rejestrów i ponownie uruchom Excela.



Jeżeli masz problemy z uruchomieniem Excela, przyczyną może być uszkodzony klucz rejestrów. Możesz spróbować przywrócić funkcjonalność Excela, uruchamiając Edytor rejestru i usuwając całą sekcję Excela:

```
HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Excel
```

Po ponownym uruchomieniu Excela wszystkie niezbędne klucze rejestrów zostaną przebudowane i Excel powinien się uruchomić. Oczywiście po wykonaniu takiej operacji utracisz wszystkie przechowywane w rejestrze informacje na temat dostosowania Excela do Twoich potrzeb.