

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

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

Excel 2003 PL. Programowanie w VBA. Vademecum profesjonalisty

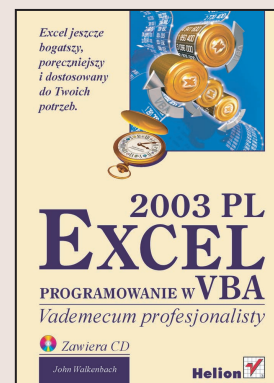
Autor: John Walkenbach

Tłumaczenie: Radosław Meryk, Piotr Pilch

ISBN: 83-7361-504-0

Tytuł oryginału: [Excel 2003 Power Programming with VBA](#)

Format: B5, stron: 872



Excel jeszcze bogatszy, poręczniejszy i dostosowany do Twoich potrzeb

Możliwości Excela są ogromne, jednak nie zawsze w pełni odpowiadają naszym potrzebom. Najpopularniejszy obecnie arkusz kalkulacyjny zawiera także wiele interesujących funkcji ukrytych w tle i nieznanymi typowemu użytkownikowi. Czy istnieje sposób na wykorzystanie ukrytych funkcji Excela oraz dostosowanie tych znanych do wymagań stawianych przez konkretne zadanie? Tak – tym sposobem jest Visual Basic for Applications (VBA). VBA to język programowania umożliwiający tworzenie aplikacji bazujących na możliwościach innych aplikacji, w tym przypadku – na możliwościach Excela.

Książka „Excel 2003 PL. Programowanie w VBA. Vademecum profesjonalisty” przeznaczona jest dla wszystkich użytkowników Excela, którzy chcą pogłębić swoją wiedzę o tej aplikacji i stworzyć własne, powiązane z nią, programy. Przedstawia podstawy programowania w Excelu oraz techniki zaawansowane – wszystko, co jest związane z projektowaniem aplikacji w Excelu i programowaniem w języku VBA.

- Niezbędne wiadomości o Excelu i wykorzystywanych przez niego formatach plików
- Zasady projektowania aplikacji arkusza kalkulacyjnego
- Programowanie w języku VBA
- Tworzenie niestandardowych okien dialogowych – formularzy UserForm
- Pisanie narzędzi dla programu Excel
- Praca z wykresami
- Programowanie obsługi zdarzeń
- Współpraca z innymi aplikacjami
- Tworzenie elementów aplikacji – menu, pasków narzędzi i systemów pomocy
- Operacje na plikach
- Komponenty języka VBA

Niewielu użytkowników Excela jest tak naprawdę świadomych jego możliwości. Przeczytaj niniejszą książkę i dołącz do tej elitarnej grupy.



Spis treści

O Autorze	19
Przedmowa	21
Część I	Podstawowe informacje.....27
Rozdział 1.	Skąd się wziął Excel 2003?..... 29
Historia arkuszy kalkulacyjnych w zarysie	29
Wszystko zaczęło się od programu VisiCalc	29
Program 1-2-3 firmy Lotus.....	30
Program Quattro Pro.....	33
Program Microsoft Excel.....	34
Excel jako dobre narzędzie dla projektantów aplikacji	36
Rola Excela w strategii Microsoftu	38
Rozdział 2.	Program Excel w zarysie 39
Myślenie obiektowe	39
Skoroszyty	40
Arkusze	40
Arkusze wykresów	41
Arkusze makr XLM.....	42
Arkusze dialogowe programów Excel 5/95.....	42
Interfejs użytkownika w Excelu	42
Menu	42
Menu podręczne	43
Paski narzędzi	43
Okna dialogowe.....	44
Funkcja „przeciągnij i upuść”.....	45
Skróty klawiaturowe.....	45
Inteligentne tagi	45
Panel zadań	46
Wprowadzanie danych.....	46
Formuły, funkcje i nazwy	48
Dostosowywanie zawartości okna Excela	49
Zaznaczanie obiektów.....	49
Formatowanie	50
Formatowanie liczbowe.....	50
Formatowane style	50
Kształty	51
Wykresy	52
Makra i programowanie.....	52

Dostęp do bazy danych	53
Arkuszwowe bazy danych.....	53
Zewnętrzne bazy danych	55
Funkcje internetowe.....	55
Funkcje związane z językiem XML	56
Narzędzia analizy.....	56
Konspekty.....	56
Automatyczne sumy częściowe.....	56
Dodatek Analysis ToolPak	56
Tabele przestawne	57
Dodatek Solver	58
Dodatki.....	58
Zgodność wersji	58
Opcje ochrony.....	58
Ochrona formuł przed nadpisaniem	59
Ochrona struktury skoroszytu.....	59
Ochrona skoroszytu przy użyciu hasła	60
Ochrona kodu języka VBA przy użyciu hasła.....	60
System pomocy Excela.....	61
Rozdział 3. Wybrane zasady stosowania formuł	63
Formuły.....	63
Obliczanie formuł	63
Odwołania do komórki lub zakresu	64
Dlaczego warto stosować odwołania, które nie są względne?	65
Notacja WIK1	65
Odwołania do innych arkuszy lub skoroszytów	66
Zastosowanie nazw	67
Nadawanie nazw komórkom i zakresom.....	67
Nadawanie nazw istniejącym odwołaniom	68
Stosowanie nazw z operatorem przecięcia	69
Nadawanie nazw kolumnom i wierszom.....	70
Obszar obowiązywania nazw	70
Nadawanie nazw stałym.....	70
Nadawanie nazw formułom.....	71
Nadawanie nazw obiektom.....	73
Błędy formuł.....	73
Formuły tablicowe	74
Przykład formuły tablicowej	74
Kalendarz oparty na formule tablicowej	75
Zalety i wady formuły tablicowej.....	75
Metody zliczania i sumowania	76
Zastosowanie funkcji LICZ.JEŻELI lub SUMA.JEŻELI.....	76
Zastosowanie formuł tablicowych do zliczania i sumowania	77
Inne narzędzia zliczające	79
Używanie daty i czasu	79
Wprowadzanie daty i czasu	79
Stosowanie dat sprzed roku 1900	80
Tworzenie megaformuł.....	81
Rozdział 4. Pliki Excela	85
Uruchamianie Excela.....	85
Obsługiwane formaty plików arkuszy kalkulacyjnych.....	87
Pliki arkusza kalkulacyjnego 1-2-3 firmy Lotus	87
Pliki arkusza kalkulacyjnego Quattro Pro	88

Formaty plików baz danych	88
Formaty plików tekstowych	89
Inne formaty plików	89
Pliki tworzone przez Excel	89
Pliki formatu XLS	90
Pliki obszaru roboczego.....	90
Pliki szablonów.....	91
Pliki pasków narzędzi.....	91
Pliki dodatków	92
Excel i język HTML	92
W jaki sposób Excel korzysta z formatu HTML?	93
Zwiększanie złożoności.....	94
Tworzenie interaktywnych plików HTML.....	94
Importowanie i eksportowanie plików XML.....	96
Czym jest XML?	96
Importowanie zawartości plików XML przy użyciu mapy	97
Importowanie zawartości plików XML do listy.....	99
Eksportowanie zawartości plików XML z Excela	99
Ustawienia Excela w rejestrze systemu Windows.....	100
Rejestr systemu Windows	100
Ustawienia Excela	101

Część II Projektowanie aplikacji w Excelu..... 103

Rozdział 5. Czym jest aplikacja arkusza kalkulacyjnego?..... 105

Robocza definicja aplikacji arkusza kalkulacyjnego	105
Projektant i użytkownik końcowy	106
Kim są projektanci i czym się zajmują?	107
Klasyfikacja użytkowników arkuszy kalkulacyjnych	108
Odbiorcy aplikacji arkusza kalkulacyjnego.....	108
Rozwiązywanie problemów przy użyciu aplikacji arkusza kalkulacyjnego.....	109
Podstawowe kategorie aplikacji arkusza kalkulacyjnego	110
Arkusze kalkulacyjne tworzone szybko i niestarczanie	111
Arkusze kalkulacyjne przeznaczone wyłącznie do użytku prywatnego.....	111
Aplikacje jednego użytkownika	112
Aplikacje typu „spaghetti”.....	112
Aplikacje użytkowe	112
Dodatki zawierające funkcje arkusza	113
Jednoblokowe budżety	113
Modele warunkowe	114
Aplikacje przechowujące dane i udzielające do nich dostępu.....	114
Aplikacje komunikujące się z bazami danych.....	115
Aplikacje „pod klucz”	115

Rozdział 6. Podstawy projektowania aplikacji arkusza kalkulacyjnego

Podstawowe etapy projektowania.....	117
Określenie wymagań użytkownika	118
Planowanie aplikacji spełniającej wymagania użytkownika	119
Określenie najważniejszego interfejsu użytkownika	121
Tworzenie niestandardowych okien dialogowych	122
Zastosowanie kontrolek ActiveX w arkuszu	122
Dostosowanie menu.....	123
Dostosowywanie pasków narzędzi.....	125
Tworzenie skrótów klawiaturowych	127
Rozpoczęcie prac projektowych.....	127

Zadania realizowane z myślą o końcowym użytkowniku	127
Testowanie aplikacji	128
Zwiększanie odporności aplikacji	128
Tworzenie aplikacji intuicyjnej i estetycznie wyglądającej	131
Tworzenie systemu pomocy przeznaczonego dla użytkownika	132
Dokumentowanie prac projektowych	132
Przekazanie aplikacji użytkownikom	132
Uaktualnianie aplikacji w razie konieczności	133
Pozostałe kwestie dotyczące projektowania	134
Wersja Excela zainstalowana przez użytkownika	134
Wersja językowa	134
Szybkość systemu	135
Tryby karty graficznej	135

Część III Język Visual Basic for Applications 137

Rozdział 7. Wprowadzenie do języka VBA 139

Podstawowe informacje o języku BASIC	139
Język VBA	139
Modele obiektowe	140
Porównanie języka VBA z językiem XLM	140
Wprowadzenie do języka VBA	140
Edytor Visual Basic	144
Uruchomienie edytora VBE	144
Okna edytora Visual Basic	145
Zastosowanie okna Project Explorer	146
Dodanie nowego modułu VBA	147
Usuwanie modułu VBA	148
Eksportowanie i importowanie obiektów	148
Zastosowanie okien Code	148
Minimalizacja i maksymalizacja okien	149
Przechowywanie kodu źródłowego języka VBA	149
Wprowadzanie kodu źródłowego języka VBA	150
Dostosowywanie edytora Visual Basic	155
Zakładka Editor	156
Zakładka Editor Format	158
Zakładka General	159
Zastosowanie zakładki Docking	160
Rejestrator makr Excela	160
Co właściwie rejestrator makr zapisuje?	161
Względne czy bezwzględne?	162
Opcje związane z rejestrowaniem	165
Modyfikowanie zarejestrowanych makr	166
Obiekty i zbiory	167
Hierarchia obiektów	168
Zbiory	168
Odwoływanie się do obiektów	169
Właściwości i metody	169
Właściwości obiektów	170
Metody obiektów	171
Obiekt Comment	172
Pomoc dotycząca obiektu Comment	172
Właściwości obiektu Comment	173
Metody obiektu Comment	173

	Zbiór Comments	174
	Właściwość Comment	175
	Obiekty podlegające obiektowi Comment	175
	Sprawdzanie, czy komórka zawiera komentarz	177
	Dodanie nowego obiektu Comment	177
	Kilka przydatnych właściwości obiektu Application	178
	Obiekty Range	179
	Właściwość Range	180
	Właściwość Cells	181
	Właściwość Offset	182
	Co należy wiedzieć o obiektach?	184
	Podstawowe zagadnienia, które należy zapamiętać	184
	Dodatkowe informacje na temat obiektów i właściwości	185
Rozdział 8.	Podstawy programowania w języku VBA	189
	Przegląd elementów języka VBA	189
	Komentarze	191
	Zmienne, typy danych i stałe	192
	Definiowanie typów danych	193
	Deklarowanie zmiennych	196
	Zasięg zmiennych	197
	Zastosowanie stałych	201
	Zastosowanie łańcuchów	202
	Zastosowanie dat	203
	Instrukcje przypisania	204
	Tablice	205
	Deklarowanie tablic	205
	Deklarowanie tablic wielowymiarowych	206
	Zmienne obiektowe	207
	Typy danych definiowane przez użytkownika	208
	Funkcje wbudowane	208
	Manipulowanie obiektami i zbiorami	211
	Konstrukcja With ... End With	211
	Konstrukcja For Each ... Next	212
	Sterowanie wykonywaniem procedur	214
	Instrukcja GoTo	214
	Konstrukcja If ... Then	215
	Konstrukcja Select Case	218
	Wykonywanie bloku instrukcji w ramach pętli	221
Rozdział 9.	Zastosowanie procedur Sub języka VBA	227
	Deklarowanie i tworzenie procedur Sub języka VBA	227
	Deklarowanie procedury Sub	227
	Zasięg procedury	228
	Wykonywanie procedur Sub	229
	Wykonywanie procedury przy użyciu polecenia Run Sub/UserForm	230
	Uruchamianie procedury z poziomu okna dialogowego Makro	230
	Wykonywanie procedury przy użyciu skrótu opartego na klawiszu Ctrl	231
	Wykonywanie procedury przy użyciu niestandardowego menu	232
	Wywoływanie procedury z innej procedury	233
	Wykonywanie procedury przy użyciu przycisku paska narzędzi	237
	Wykonywanie procedury poprzez kliknięcie obiektu	238
	Wykonywanie makra po wystąpieniu zdarzenia	239
	Wykonywanie procedury z poziomu okna Immediate	239

Przekazywanie argumentów procedurom	240
Metody obsługi błędów	243
Przechwytywanie błędów	243
Przykłady kodu źródłowego obsługującego błędy	245
Rzeczywisty przykład wykorzystujący procedury Sub	247
Cel.....	247
Wymagania projektowe.....	247
Dostępne informacje.....	248
Sposób realizacji.....	248
Co należy wiedzieć?.....	249
Wstępne rejestrowanie makra.....	249
Wstępne przygotowania	250
Pisanie kodu źródłowego.....	251
Tworzenie procedury sortującej	252
Dodatkowe testy	255
Usuwanie problemów	256
Dostępność narzędzia	258
Ocenianie projektu.....	259
Rozdział 10. Tworzenie procedur Function.....	261
Porównanie procedur Sub i Function.....	261
Dlaczego tworzy się funkcje niestandardowe?	261
Pierwszy przykład procedury Function	262
Funkcja niestandardowa	262
Zastosowanie funkcji w arkuszu.....	263
Zastosowanie funkcji w procedurze języka VBA	264
Analiza funkcji niestandardowej	264
Procedury Function.....	265
Deklarowanie funkcji	265
Zasięg funkcji	266
Wykonywanie procedur Function	267
Argumenty procedury Function.....	268
Przykłady funkcji	268
Funkcja pozbawiona argumentów	269
Kolejna funkcja pozbawiona argumentów	270
Funkcja z jednym argumentem.....	270
Funkcja z dwoma argumentami.....	273
Funkcja pobierająca tablicę jako argument	274
Funkcja używająca opcjonalnych argumentów	274
Funkcja zwracająca tablicę języka VBA	276
Funkcja zwracająca wartość błędu	278
Funkcja o nieokreślonej liczbie argumentów	280
Emulowanie funkcji SUMA Excela	280
Funkcje wykrywające i usuwające błędy.....	283
Okno dialogowe Wstawianie funkcji.....	284
Definiowanie kategorii funkcji.....	285
Dodanie opisu funkcji.....	286
Dodatki przechowujące funkcje niestandardowe	287
Interfejs API systemu Windows	287
Przykłady zastosowania funkcji interfejsu API systemu Windows	288
Identyfikacja katalogu systemu Windows	288
Wykrywanie wciśnięcia klawisza Shift.....	289
Dodatkowe informacje na temat funkcji interfejsu API.....	290

Rozdział 11. Przykłady i metody programowania w języku VBA	291
Przetwarzanie zakresów	291
Kopiowanie zakresu	292
Przenoszenie zakresu	293
Kopiowanie zakresu o zmiennej wielkości	293
Zaznaczanie różnego typu zakresów i identyfikowanie ich	294
Wprowadzanie wartości do komórki	296
Wprowadzanie wartości do następnej pustej komórki	297
Wstrzymywanie wykonywania makra w celu pobrania zakresu zaznaczonego przez użytkownika	298
Zliczanie zaznaczonych komórek	299
Określanie typu zaznaczonego zakresu	300
Wydajne przetwarzanie komórek zaznaczonego zakresu przy użyciu pętli	302
Usuwanie wszystkich pustych wierszy	303
Określanie, czy zakres zawiera się w innym zakresie	304
Określanie typu danych zawartych w komórce	304
Odczytywanie i zapisywanie zakresów	305
Lepsza metoda zapisywania zakresu	306
Przenoszenie zawartości tablic jednowymiarowych	308
Przenoszenie zawartości zakresu do tablicy typu Variant	308
Zaznaczanie maksymalnej wartości zakresu	309
Zaznaczanie wszystkich komórek określonego formatu	310
Przetwarzanie skoroszytów i arkuszy	312
Zapisywanie wszystkich skoroszytów	312
Zapisywanie i zamykanie wszystkich skoroszytów	312
Korzystanie z właściwości skoroszytu	312
Synchronizowanie arkuszy	313
Metody programowania w języku VBA	314
Przełączanie wartości właściwości typu logicznego	314
Określanie liczby drukowanych stron	315
Wyświetlanie daty i czasu	315
Pobieranie listy czcionek	317
Sortowanie tablicy	318
Przetwarzanie grupy plików	319
Funkcje przydatne w procedurach języka VBA	320
Funkcja FileExists	321
Funkcja FileNameOnly	321
Funkcja PathExists	321
Funkcja RangeNameExists	322
Funkcja SheetExists	322
Funkcja WorkbookIsOpen	322
Pobieranie wartości z zamkniętego skoroszytu	322
Funkcje przydatne w formułach arkusza	324
Funkcje zwracające informacje o formatowaniu komórki	324
Wyświetlanie daty w trakcie zapisywania lub drukowania pliku	325
Obiekty nadrzędne	326
Zliczanie komórek, których wartości zawierają się pomiędzy dwoma wartościami	327
Zliczanie widocznych komórek zakresu	328
Określanie ostatniej niepustej komórki kolumny lub wiersza	328
Czy łańcuch jest zgodny z wzorcem?	330
Wydzielanie n-tego elementu łańcucha	331
Funkcja wielofunkcyjna	332
Funkcja SHEETOFFSET	333
Zwracanie maksymalnej wartości ze wszystkich arkuszy	334

Zwracanie tablicy zawierającej unikatowe, losowo uporządkowane liczby całkowite	335
Porządkowanie zakresu w losowy sposób	336
Funkcje interfejsu API systemu Windows	338
Określanie skojarzeń plików	338
Określenie informacji dotyczących domyślnej drukarki	339
Określenie aktualnej rozdzielczości karty graficznej	340
Dodanie dźwięku do aplikacji	340
Odczytywanie zawartości rejestru systemu Windows i zapisywanie w nim danych	342

Część IV Zastosowanie formularzy UserForm 345

Rozdział 12. Alternatywne metody tworzenia niestandardowych okien dialogowych 347

Okno wprowadzania danych	347
Funkcja InputBox języka VBA	347
Metoda InputBox Excela	349
Okno komunikatu — funkcja MsgBox języka VBA	351
Metoda GetOpenFilename Excela	354
Metoda GetSaveAsFilename Excela	357
Okno wybierania katalogu	357
Wybieranie katalogu przy użyciu funkcji interfejsu API systemu Windows	358
Wybieranie katalogu przy użyciu obiektu FileDialog	360
Wyświetlanie wbudowanych okien dialogowych Excela	360
Zastosowanie zbioru Dialogs	361
Dodatkowe informacje na temat wbudowanych okien dialogowych	362
Zastosowanie argumentów z wbudowanymi oknami dialogowymi	363
Bezpośrednie wybieranie pozycji menu	363

Rozdział 13. Wprowadzenie do formularzy UserForm 365

Wstawianie nowego formularza UserForm	365
Dodawanie kontrolki do formularza UserForm	366
Kontrolki okna Toolbox	367
Kontrolka CheckBox	367
Kontrolka ComboBox	367
Kontrolka CommandButton	367
Kontrolka Frame	367
Kontrolka Image	369
Kontrolka Label	369
Kontrolka ListBox	369
Kontrolka MultiPage	369
Kontrolka OptionButton	369
Kontrolka RefEdit	370
Kontrolka ScrollBar	370
Kontrolka SpinButton	370
Kontrolka TabStrip	370
Kontrolka TextBox	370
Kontrolka ToggleButton	370
Modyfikowanie kontrolki formularza UserForm	370
Modyfikowanie właściwości kontrolki	371
Zastosowanie okna Properties	372
Wspólne właściwości	373
Zdobywanie dodatkowych informacji o właściwościach	373
Uwzględnienie wymagań użytkowników preferujących korzystanie z klawiatury	373

Wyświetlanie i zamykanie formularzy UserForm	375
Wyświetlanie formularza UserForm	375
Zamykanie formularza UserForm	377
Procedury obsługi zdarzeń.....	378
Przykład tworzenia formularza UserForm.....	378
Tworzenie formularza UserForm	378
Pisanie kodu źródłowego procedury wyświetlającej okno dialogowe	381
Testowanie okna dialogowego	381
Dodawanie procedur obsługi zdarzeń.....	382
Sprawdzanie poprawności danych.....	384
Ukończenie tworzenia okna dialogowego	384
Zdarzenia powiązane z formularzem UserForm.....	385
Zdobywanie informacji na temat zdarzeń	385
Zdarzenia formularza UserForm	386
Zdarzenia związane z kontrolką SpinButton	386
Współpraca kontrolki SpinButton z kontrolką TextBox	388
Odwoływanie się do kontrolki formularza UserForm	390
Dostosowywanie okna Toolbox do własnych wymagań	391
Modyfikacja ikon lub tekstu podpowiedzi	391
Dodawanie nowych zakładek	391
Dostosowywanie lub łączenie kontrolki	392
Dodawanie nowych kontrolki ActiveX	392
Tworzenie szablonów formularzy UserForm	393
Lista kontrolna związana z tworzeniem formularzy UserForm.....	394

Rozdział 14. Przykłady formularzy UserForm..... 395

Tworzenie formularza UserForm pełniącego funkcję menu	395
Zastosowanie w formularzu UserForm kontrolki CommandButton	395
Zastosowanie w formularzu UserForm kontrolki ListBox.....	396
Zaznaczanie zakresów przy użyciu formularza UserForm	397
Tworzenie okna powitalnego.....	398
Wyłączanie przycisku Zamknij formularza UserForm.....	400
Zmiana wielkości formularza UserForm	401
Powiększanie i przewijanie arkusza przy użyciu formularza UserForm	402
Zastosowania kontrolki ListBox	404
Kontrolka ListBox	404
Umieszczanie pozycji w kontrolce ListBox	405
Identyfikowanie zaznaczonej pozycji.....	408
Identyfikowanie wielu zaznaczonych pozycji kontrolki ListBox	409
Wiele list w jednej kontrolce ListBox	410
Przenoszenie pozycji kontrolki ListBox.....	410
Przemieszczanie pozycji kontrolki ListBox	412
Stosowanie wielokolumnowych kontrolki ListBox.....	413
Zastosowanie kontrolki ListBox do wybierania wierszy arkusza	415
Uaktywnianie arkusza za pomocą kontrolki ListBox.....	417
Zastosowania kontrolki MultiPage	418

Rozdział 15. Zaawansowane techniki korzystania z formularzy UserForm..... 421

Wyświetlanie wskaźnika postępu zadania	421
Tworzenie samodzielnego wskaźnika postępu zadania.....	422
Wyświetlanie wskaźnika postępu zadania za pomocą kontrolki MultiPage	424
Wyświetlanie wskaźnika postępu zadania bez korzystania z kontrolki MultiPage.....	426
Kreatory — interaktywne sekwencje okien dialogowych	427
Konfigurowanie kontrolki MultiPage w celu utworzenia kreatora	428
Dodawanie przycisków do formularza UserForm kreatora.....	428

Programowanie przycisków kreatora	429
Zależności programowe w kreatorach	430
Wykonywanie zadań za pomocą kreatorów	431
Emulacja funkcji MsgBox	432
Emulacja funkcji MsgBox: kod funkcji MsgBox	433
Jak działa funkcja emulująca MsgBox	434
Wykorzystanie funkcji MsgBox	435
Niemodalne okna dialogowe	435
Obsługa wielu przycisków formularza UserForm za pomocą jednej procedury obsługi zdarzeń	438
Wybór koloru za pomocą formularza UserForm	441
Wyświetlanie wykresów w formularzach UserForm	442
Metoda 1. zapisanie wykresu do pliku	443
Metoda 2. zastosowanie kontrolki ChartSpace z pakietu OWC	444
Wyświetlanie arkuszy w formularzach UserForm	446
Udostępnianie kontrolki Spreadsheet	447
Dodawanie kontrolki Spreadsheet w formularzu UserForm	447
Prosty przykład zastosowania kontrolki Spreadsheet komponentów sieci WWW pakietu Office	447
UserForm Deluxe: ulepszony formularz danych	450
Opis ulepszanego formularza danych	450
Instalacja dodatku — ulepszanego formularza danych	450
Wykorzystanie ulepszanego formularza danych	451

Część V Zaawansowane techniki programowania453

Rozdział 16. Tworzenie narzędzi dla Excela w języku VBA 455

Wprowadzenie	455
Zastosowanie języka VBA do tworzenia narzędzi	456
Co decyduje o przydatności narzędzia?	456
Operacje tekstowe — anatomia narzędzia	457
Podstawy tworzenia narzędzia Operacje tekstowe	458
Określenie wymagań dla narzędzia Operacje tekstowe	458
Jak działa narzędzie Operacje tekstowe?	458
Skoroszyt narzędzia Operacje tekstowe	459
Formularz UserForm dla narzędzia Operacje tekstowe	459
Moduł kodu ThisWorkbook	461
Moduł VBA Module1	462
Moduł formularza UserForm1	463
Poprawa wydajności narzędzia Operacje tekstowe	464
Zapisywanie ustawień narzędzia Operacje tekstowe	465
Implementacja procedury Cofnij	466
Ocena realizacji projektu	468
Działanie narzędzia Operacje tekstowe	469
Dodatkowe informacje na temat narzędzi Excela	469

Rozdział 17. Tabele przestawne 471

Wykorzystanie języka VBA do tworzenia tabel przestawnych	471
Tworzenie tabel przestawnych	471
Analiza zarejestrowanego kodu tworzenia tabeli przestawnej	473
Ulepszanie zarejestrowanego kodu tworzenia tabeli przestawnej	473
Złożone tabele przestawne	474
Dane dla złożonej tabeli przestawnej	475
Kod tworzący tabelę przestawną	475
Jak działa złożona tabela przestawna?	477

Tworzenie tabel przestawnych na podstawie zewnętrznej bazy danych	478
Jednoczesne tworzenie wielu tabel przestawnych	479
Modyfikowanie tabel przestawnych	482
Rozdział 18. Wykresy	485
Podstawowe wiadomości o wykresach	485
Położenie wykresu	485
Model obiektu Chart	486
Rejestrowanie makr dotyczących wykresów	487
Zarejestrowany kod makra tworzącego wykres	487
Ulepszanie zarejestrowanego kodu tworzenia wykresu	489
Popularne techniki języka VBA dotyczące wykresów	489
Wykorzystanie VBA do uaktywnienia wykresu	490
Wykorzystanie VBA do deaktywacji wykresu	491
Sprawdzanie, czy wykres uaktywniono	492
Usuwanie elementów z kolekcji ChartObjects lub Charts	492
Formatowanie wykresów za pomocą VBA	493
Przetwarzanie wszystkich wykresów w pętli	494
Zmiana rozmiarów i wyrównywanie obiektów ChartObject	494
Inne techniki przetwarzania wykresów	496
Zastosowanie nazw w formule SERIE	496
Zastosowanie języka VBA w celu określenia danych	
wykorzystywanych na wykresie	497
Zastosowanie języka VBA w celu określenia zakresu danych	
wykorzystywanych na wykresie	500
Wykorzystanie VBA do wyświetlania dowolnych etykiet danych na wykresie	503
Wyświetlanie wykresu w oknie UserForm	505
Zdarzenia związane z wykresami	507
Przykład wykorzystania zdarzeń związanych z wykresami	507
Obsługa zdarzeń dla wykresów wbudowanych	511
Zastosowanie zdarzeń dla wykresów wbudowanych	512
Jak ułatwić sobie pracę z wykresami?	514
Drukowanie wbudowanych wykresów	515
Tworzenie „martwych” wykresów	515
Wykorzystanie zdarzenia MouseOver do wyświetlania tekstu	517
Wykresy animowane	518
Tworzenie wykresu krzywych hipocykloidalnych	519
Tworzenie wykresu-zegara	520
Co można zrobić z wykresami bez użycia makr?	522
Sterowanie seriami danych za pomocą automatycznego filtrowania	522
Zapisywanie wielu wykresów w arkuszu-wykresie	523
Tworzenie samorozszerzającego się wykresu	524
Tworzenie interaktywnego wykresu	530
Rozdział 19. Obsługa zdarzeń	535
Typy zdarzeń, które można monitorować w Excelu	535
Najważniejsze informacje o zdarzeniach	536
Sekwencje zdarzeń	536
Gdzie należy umieścić procedury obsługi zdarzeń?	537
Wyłączanie obsługi zdarzeń	538
Wprowadzanie kodu procedury obsługi zdarzeń	539
Procedury obsługi zdarzeń z argumentami	540
Zdarzenia poziomu skoroszytu	541
Zdarzenie Open	541
Zdarzenie Activate	543

Zdarzenie SheetActivate.....	543
Zdarzenie NewSheet.....	544
Zdarzenie BeforeSave.....	544
Zdarzenie Deactivate.....	544
Zdarzenie BeforePrint.....	545
Zdarzenie BeforeClose.....	546
Zdarzenia poziomu arkusza.....	547
Zdarzenie Change.....	547
Monitorowanie modyfikacji w wybranym zakresie komórek.....	549
Zdarzenie SelectionChange.....	551
Zdarzenie BeforeRightClick.....	552
Zdarzenia dotyczące wykresów.....	553
Zdarzenia dotyczące aplikacji.....	553
Włączenie obsługi zdarzeń poziomu aplikacji.....	554
Sprawdzanie, czy skoroszyt jest otwarty.....	556
Monitorowanie zdarzeń poziomu aplikacji.....	557
Zdarzenia dotyczące formularzy UserForm.....	558
Zdarzenia niezwiązane z obiektami.....	559
Zdarzenie OnTime.....	560
Zdarzenie OnKey.....	561
Rozdział 20. Interakcje z innymi aplikacjami.....	563
Uruchamianie innych aplikacji z poziomu Excela.....	563
Zastosowanie funkcji Shell języka VBA.....	563
Zastosowanie funkcji Windows API ShellExecute.....	566
Uaktywnianie aplikacji z poziomu Excela.....	567
Wykorzystanie instrukcji AppActivate.....	567
Uaktywnianie aplikacji pakietu Microsoft Office.....	567
Uruchamianie okien dialogowych Panelu sterowania.....	568
Wykorzystanie automatyzacji.....	569
Działania z obiektami innych aplikacji z wykorzystaniem automatyzacji.....	569
Wczesne i późne wiązanie.....	570
Prosty przykład późnego wiązania.....	572
Zarządzanie Wordem z poziomu Excela.....	573
Zarządzanie Excelem z poziomu innej aplikacji.....	576
Wysyłanie spersonalizowanych wiadomości e-mail z wykorzystaniem Outlooka.....	577
Działania z obiektami danych ActiveX (ADO).....	579
Wysyłanie wiadomości e-mail z załącznikami z poziomu Excela.....	581
Zastosowanie metody SendKeys.....	582
Rozdział 21. Tworzenie i wykorzystanie dodatków.....	585
Czym są dodatki?.....	585
Porównanie dodatku ze standardowym skoroszytem.....	585
Po co tworzy się dodatki?.....	586
Menedżer dodatków Excela.....	587
Tworzenie dodatków.....	588
Przykładowy dodatek.....	589
Konfiguracja skoroszytu dla przykładowego dodatku.....	589
Testowanie skoroszytu użytego do utworzenia przykładowego dodatku.....	589
Wprowadzanie opisu dla przykładowego dodatku.....	590
Utworzenie dodatku.....	590
Instalowanie dodatku.....	592
Dystrybucja dodatków.....	592
Modyfikowanie dodatku.....	592

Porównanie plików XLA i XLS	594
Rozmiar i struktura plików XLS i XLA	594
Pliki XLA — przynależność do kolekcji z poziomu VBA	594
Widoczność plików XLS i XLA	594
Arkusze i wykresy w plikach XLS i XLA	595
Dostęp do procedur VBA w dodatku	596
Przykłady przetwarzania dodatków za pomocą kodu VBA	597
Kolekcja AddIns	598
Właściwości obiektu AddIn	599
Zdarzenia związane z obiektami Addin	602
Optymalizacja wydajności dodatków	602
Maksymalizacja szybkości kodu dodatków	602
Kontrolowanie rozmiaru pliku dodatku	603
Problemy z dodatkami	604
Zapewnienie zainstalowania dodatku	604
Odwoływanie się do innych plików z poziomu dodatku	605
Wykrywanie właściwej wersji Excela dla dodatku	605

Część VI Tworzenie aplikacji 607

Rozdział 22. Tworzenie pasków narzędzi 609

Paski narzędzi	609
Działania z paskami narzędzi	609
Jak Excel zarządza paskami narzędzi?	610
Przechowywanie pasków narzędzi	610
Problemy z działaniem pasków narzędzi	611
Ręczne dostosowywanie pasków narzędzi	611
Tryb dostosowywania pasków narzędzi	612
Dystrybucja pasków narzędzi użytkownika	615
Działania z kolekcją CommandBars	617
Rodzaje pasków narzędzi	617
Wyświetlenie wszystkich obiektów CommandBar	617
Tworzenie pasków narzędzi	618
Odwoływanie się do pasków narzędzi w VBA	619
Usuwanie paska narzędzi za pomocą kodu VBA	620
Właściwości pasków narzędzi	620
Odwoływanie się do kontrolki na pasku narzędzi	625
Wyszczególnienie kontrolki na pasku narzędzi	626
Wyświetlanie wszystkich kontrolki na wszystkich paskach narzędzi	627
Dodawanie kontrolki na pasku narzędzi	628
Usuwanie kontrolki z paska narzędzi	628
Właściwości kontrolki pasków narzędzi	629

Rozdział 23. Tworzenie menu 639

Pasek menu w Excelu	639
Modyfikowanie menu Excela przez użytkownika	640
Pojęcia związane z systemem menu Excela	640
Usuwanie elementów menu Excela	641
Dodawanie elementów menu Excela	641
Modyfikacja elementów menu Excela	642
Wykorzystanie języka VBA do dostosowywania menu w Excelu	642
Wyświetlanie informacji o menu Excela	643
Dodawanie nowego menu na pasku menu	644
Usuwanie menu z paska menu	646
Dodawanie pozycji menu	647

Wyświetlanie klawisza skrótu wraz z nazwą polecenia	650
Otwieranie menu, które zostało usunięte	651
Wykorzystanie zdarzeń do programowania menu	652
Automatyczne dodawanie i usuwanie menu	652
Dezaktywacja lub ukrywanie menu	653
Działania z poleceniami menu powiązаныmi z polami wyboru	654
Tworzenie menu użytkownika — łatwy sposób	657
Utworzenie zastępczego paska menu arkusza kalkulacyjnego	659
Operacje z menu podręcznymi	660
Dodawanie pozycji do menu podręcznych	662
Usuwanie pozycji z menu podręcznych	663
Dezaktywacja pozycji menu podręcznych	663
Dezaktywacja menu podręcznych	663
Przywracanie ustawień menu podręcznych	664
Tworzenie nowych menu podręcznych	664
Rozdział 24. Tworzenie systemów pomocy w aplikacjach.....	667
Dlaczego należy tworzyć systemy pomocy w aplikacjach?	667
Systemy pomocy wykorzystujące komponenty Excela	668
Wykorzystanie komentarzy w celu tworzenia systemów pomocy	668
Wykorzystanie pól tekstowych w celu utworzenia systemu pomocy	670
Wykorzystanie arkusza do wyświetlania tekstu pomocy	670
Wyświetlanie pomocy w oknie UserForm	671
Wykorzystanie asystenta pakietu Office do wyświetlania pomocy	674
Symulacja właściwości Co to jest? za pomocą formularza UserForm	676
Wykorzystanie systemu HTML Help	677
Powiązanie pliku pomocy z aplikacją	678
Wiązanie tematów pomocy z funkcjami VBA	679
Inne sposoby wyświetlania plików pomocy HTML Help	680
Wykorzystanie metody Help	680
Wyświetlanie pomocy z okna informacyjnego	680
Wyświetlanie pomocy z okna InputBox	681
Rozdział 25. Tworzenie aplikacji wygodnych dla użytkownika	683
Co to jest aplikacja user-oriented?	683
Przykładowa aplikacja — Kreator amortyzacji pożyczek	683
Obsługa Kreatora amortyzacji pożyczek	683
Struktura skoroszytu Kreatora amortyzacji pożyczek	685
Jak działa Kreator amortyzacji pożyczek?	686
Potencjalne usprawnienia Kreatora amortyzacji pożyczek	690
Wskazówki dotyczące projektowania aplikacji	690
Część VII Inne zagadnienia	693
Rozdział 26. Problemy zgodności	695
Co to jest zgodność?	695
Rodzaje problemów zgodności	696
Obsługiwane formaty plików Excela	697
Kiedy trzeba unikać nowych właściwości?	698
Czy aplikacja będzie działać na komputerach Macintosh?	698
Tworzenie aplikacji dla wielu wersji narodowych	699
Aplikacje obsługujące wiele języków	701
Obsługa języka w kodzie VBA	702
Wykorzystanie właściwości lokalnych	702
Identyfikacja ustawień systemu	703
Ustawienia daty i godziny	705

Rozdział 27. Operacje na plikach wykonywane za pomocą kodu VBA.....	707
Wykonywanie popularnych operacji na plikach.....	707
Wykorzystanie poleceń języka VBA do wykonywania operacji na plikach.....	707
Wykorzystanie obiektu FileSearch.....	710
Wykorzystanie obiektu FileSystemObject.....	711
Wyszukiwanie plików zawierających określony tekst.....	713
Operacje z plikami tekstowymi.....	714
Otwieranie plików tekstowych.....	714
Czytanie plików tekstowych.....	715
Zapisywanie danych do plików tekstowych.....	716
Uzyskanie numeru pliku.....	716
Określanie lub ustawianie pozycji w pliku.....	716
Instrukcje pozwalające na odczytywanie i zapisywanie plików.....	716
Przykłady wykonywania operacji na plikach.....	717
Importowanie danych z pliku tekstowego.....	717
Eksportowanie zakresu do pliku tekstowego.....	718
Importowanie pliku tekstowego do zakresu.....	719
Rejestrowanie wykorzystania Excela.....	720
Filtrowanie pliku tekstowego.....	721
Importowanie więcej niż 256 kolumn danych do skoroszytu.....	721
Eksportowanie zakresu do pliku HTML.....	723
Eksportowanie zakresu do pliku XLM.....	726
Rozdział 28. Operacje na komponentach języka VBA.....	729
Podstawowe informacje o środowisku IDE.....	729
Model obiektów środowiska IDE.....	730
Kolekcja VBProjects.....	731
Wyświetlanie wszystkich komponentów projektu VBA.....	733
Zastępowanie modułu uaktualnioną wersją.....	734
Wykorzystanie języka VBA do generowania kodu VBA.....	736
Wykorzystywanie kodu VBA do umieszczenia kontrolki w formularzu UserForm	
w fazie projektowania.....	738
Operacje z formularzami UserForm w fazie projektowania i wykonania.....	738
Dodanie 100 przycisków CommandButton w fazie projektowania.....	739
Programowe tworzenie formularzy UserForm.....	741
Prosty przykład formularza UserForm.....	741
Skomplikowany przykład dynamicznego formularza UserForm.....	743
Rozdział 29. Moduły klas.....	749
Czym jest moduł klasy?.....	749
Przykład: utworzenie klasy NumLock.....	750
Wstawianie modułu klasy.....	750
Dodawanie kodu VBA do modułu klasy.....	751
Wykorzystanie klasy NumLock.....	753
Dodatkowe informacje na temat modułów klas.....	754
Nadawanie nazwy klasie obiektów.....	754
Programowanie właściwości obiektów.....	754
Programowanie metod obiektów.....	756
Zdarzenia definiowane w module klasy.....	757
Przykład: klasa CSVFileClass.....	757
Zmienne poziomu modułu dla klasy CSVFileClass.....	757
Definicje właściwości klasy CSVFileClass.....	758
Definicje metod klasy CSVFileClass.....	758
Wykorzystanie obiektów CSVFileClass.....	760

Rozdział 30. Często zadawane pytania na temat programowania w Excelu	763
Pytania dotyczące Excela.....	764
Pytania dotyczące edytora Visual Basic	768
Pytania dotyczące procedur	771
Pytania dotyczące funkcji	776
Pytania dotyczące obiektów, właściwości, metod i zdarzeń.....	779
Pytania dotyczące formularzy UserForm	788
Pytania dotyczące dodatków.....	793
Pytania dotyczące pasków poleceń.....	795
Dodatki	799
Dodatek A Zasoby online dotyczące Excela	801
Pomoc techniczna Microsoftu	801
Opcje pomocy technicznej.....	801
Baza wiedzy firmy Microsoft.....	802
Macierzysta strona programu Microsoft Excel	802
Narzędzia pakietu Microsoft Office w internecie.....	802
Internetowe grupy dyskusyjne	802
Grupy dyskusyjne poświęcone arkuszom kalkulacyjnym.....	803
Grupy dyskusyjne Microsoftu	803
Wyszukiwanie informacji w grupach dyskusyjnych	804
Witryny WWW	805
Strona Spreadsheet	805
Pearson Software Consulting.....	805
Strona o Excelu Stephena Bullena.....	805
Strony o Excelu Davida McRitchie	806
Strona o Excelu Jona Peltiera	806
Mr. Excel	806
Najczęściej zadawane pytania	806
Dodatek B Instrukcje i funkcje VBA.....	807
Wywoływanie funkcji Excela w instrukcjach VBA	810
Dodatek C Kody błędów VBA	817
Dodatek D Zawartość płyty CD-ROM	821
Wymagania systemowe	821
Korzystanie z płyty CD-ROM w systemie Windows	821
Co znajduje się na płycie CD-ROM	822
Materiały utworzone przez autora tej książki.....	822
Aplikacje.....	834
Rozwiązywanie problemów	835
Skorowidz	837

Rozdział 17.

Tabele przestawne

Według niektórych użytkowników Excela tabele przestawne są jego najbardziej innowacyjną i oferującą największe możliwości właściwością. Jest to także unikatowa cecha Excela (żaden inny arkusz kalkulacyjny nią nie dysponuje). Po raz pierwszy pojawiła się w Excelu 5.

Ten rozdział nie jest wprowadzeniem w tematykę tabel przestawnych. Zakładam, że czytelnicy znają tę właściwość i potrafią ręcznie tworzyć i modyfikować tabele. Jak wiadomo, utworzenie tabeli przestawnej na podstawie danych z bazy danych lub listy umożliwia wykonywanie podsumowań danych w sposób, który bez ich zastosowania nie byłby możliwy. Poza tym jest to sposób niezwykle szybki. Dodatkowo tabele przestawne można tworzyć i modyfikować za pomocą kodu w języku VBA.



Tabele przestawne Excela znacznie ulepszono w Excelu 2000. Wykorzystano tam wydajniejsze metody buforowania danych, a także wprowadzono obsługę wykresów powiązanych z *tabelami przestawnymi*. Z tego powodu pewna część materiału zawartego w niniejszym rozdziale nie dotyczy Excela 97 lub wersji wcześniejszych.

Wykorzystanie języka VBA do tworzenia tabel przestawnych

W tym podrozdziale zaprezentowano prosty przykład wykorzystania języka VBA w celu utworzenia tabeli przestawnej. Baza danych, pokazana na rysunku 17.1, zawiera cztery pola: RepHandl, Region, Miesiąc i Sprzedaż. Każdy rekord opisuje wielkość sprzedaży jednego reprezentanta handlowego w określonym miesiącu.

Tworzenie tabel przestawnych

Na rysunku 17.2 pokazano tabelę przestawną utworzoną na podstawie danych. Wykonuje ona podsumowania sprzedaży według reprezentantów handlowych oraz miesięcy. Zawiera następujące pola:

- Region — pole strony w tabeli przestawnej;
- RepHandl — pole wiersza w tabeli przestawnej;
- Miesiąc — pole kolumny w tabeli przestawnej;
- Sprzedaż — pole danych w tabeli przestawnej, w którym wykorzystano funkcję SUMA.

Rysunek 17.1.

Z tej prostej bazy danych łatwo można utworzyć tabelę przestawną

RepHandl	Region	Miesiac	Sprzedaż
Andrzej	połnocny	Sty	33 488
Andrzej	połnocny	Lut	47 008
Andrzej	połnocny	Mar	32 128
Bogdan	połnocny	Sty	34 736
Bogdan	połnocny	Lut	92 872
Bogdan	połnocny	Mar	76 128
Czesław	południowy	Sty	41 536
Czesław	południowy	Lut	23 192
Czesław	południowy	Mar	21 736
Daniel	południowy	Sty	44 834
Daniel	południowy	Lut	32 002
Daniel	południowy	Mar	23 932

Rysunek 17.2.

Tabela przestawna utworzona na podstawie danych zamieszczonych na rysunku 17.1

Region	Sty	Lut	Mar	Suma końcowa
Andrzej	33488	47008	32128	112624
Bogdan	34736	92872	76128	203736
Czesław	41536	23192	21736	86464
Daniel	44834	32002	23932	100768
Suma końcowa	154594	195074	153924	503592

Zanim utworzyłem tę tabelę przestawną, włączyłem rejestrator makr. Wygenerowany kod zaprezentowano poniżej:

```
Sub Macro1()
    ' Zarejestrowane makro
    Range("A1").Select
    ActiveWorkbook.PivotCaches.Add _
        (SourceType:=xlDatabase, _
        SourceData:="Arkusz1!R1C1:R13C4"). _
        CreatePivotTable
    TableDestination:="", _
    TableName:="Tabela przestawna1", _
    DefaultVersion:=xlPivotTableVersion10

    ActiveSheet.PivotTableWizard _
        TableDestination:=ActiveSheet.Cells(3, 1)
    ActiveSheet.Cells(3, 1).Select

    With ActiveSheet.PivotTables("Tabela przestawna1"). _
        PivotFields("Region")
        .Orientation = xlPageField
        .Position = 1
    End With
End Sub
```

```
End With

With ActiveSheet.PivotTables("Tabela przestawna1"). _
    PivotFields("RepHandl")
    .Orientation = xlRowField
    .Position = 1
End With

With ActiveSheet.PivotTables("Tabela przestawna1"). _
    PivotFields("Miesiąc")
    .Orientation = xlColumnField
    .Position = 1
End With
ActiveSheet.PivotTables("Tabela przestawna1"). _
    AddDataField ActiveSheet.PivotTables("Tabela przestawna1"). _
    PivotFields("Sprzedaż"), "Suma z Sprzedaż", xlSum
End Sub
```

Kod wygenerowany przez rejestrator makr zależy od sposobu, w jaki utworzyliśmy tabelę przestawną. W przykładzie utworzyłem tabelę przestawną, która była pusta, dopóki nie przeciągnąłem pola z paska narzędzi *Tabela przestawna*. Alternatywną metodą jest kliknięcie przycisku *Układ* w drugim kroku kreatora *Tabela przestawna* i zdefiniowanie układu tabeli przestawnej przed jej utworzeniem.

Można oczywiście wykonać zarejestrowane makro, co spowoduje utworzenie kolejnej, identycznej tabeli przestawnej.

Analiza zarejestrowanego kodu tworzenia tabeli przestawnej

Kod języka VBA dotyczący działań z tabelami przestawnymi może się wydawać niejasny. Aby zarejestrowane makro nabrało sensu, trzeba poznać kilka obiektów (opis każdego z nich znajduje się w pomocy online):

- PivotCaches — zbiór obiektów PivotCache obiektu Workbook;
- PivotTables — zbiór obiektów PivotTable obiektu Worksheet;
- PivotFields — zbiór pól obiektu PivotTable;
- PivotItems — zbiór danych dla pola określonej kategorii;
- CreatePivotTable — metoda obiektu PivotCache tworząca tabelę przestawną na podstawie danych w buforze tabeli przestawnej;
- PivotTableWizard — metoda obiektu Worksheet tworząca tabelę przestawną. Jak się przekonamy w następnym punkcie, korzystanie z tej metody nie jest konieczne.

Ulepszanie zarejestrowanego kodu tworzenia tabeli przestawnej

Podobnie jak w przypadku większości zarejestrowanych makr, zaprezentowany wcześniej przykład nie jest tak wydajny, jak mógłby być. Można go uprościć i uczynić bardziej zrozumiałym. Kod z listingu 17.1 pozwoli utworzyć identyczną tabelę przestawną.

Listing 17.1. *Lepszy sposób generowania tabeli przestawnej za pomocą kodu VBA*

```

Sub CreatePivotTable()
    Dim PTCache As PivotCache
    Dim PT As PivotTable

    Set PTCache = ActiveWorkbook.PivotCaches.Add _
        (SourceType:=xlDatabase, _
        SourceData:=Range("A1").CurrentRegion.Address)

    Set PT = PTCache.CreatePivotTable _
        (TableDestination="", _
        TableName="Tabela przestawna1")

    With PT
        .PivotFields("Region").Orientation = xlPageField
        .PivotFields("Miesiąc").Orientation = xlColumnField
        .PivotFields("RepHandl").Orientation = xlRowField
        .PivotFields("Sprzedaż").Orientation = xlDataField
    End With
End Sub

```

Procedurę `CreatePivotTable` uproszczono (co może przyczynić się do zwiększenia jej przejrzystości), deklarując tylko dwie zmienne obiektowe: `PTCache` oraz `PT`. Zastępują one indeksowane odwołania do pól `ActiveSheet.PivotCaches` oraz `ActiveSheet.PivotTables`. Nowy obiekt `PivotCache` tworzy się za pomocą metody `Add`, a nowy obiekt `PivotTable` za pomocą metody `CreatePivotTable` obiektu `PivotCache`. W ostatniej sekcji kodu następuje dodanie pól do tabeli przestawnej i określenie lokalizacji wewnątrz niej (pola strony, kolumny, wiersza i danych).

W pierwotnie wygenerowanym makrze znajdował się zakodowany „na twardo” zakres danych służący do utworzenia obiektu `PivotCache` (tzn. `'Arkusz1!R1C1:R13C4'`). W procedurze `CreatePivotTable` tabela przestawna jest tworzona na podstawie bieżącego regionu otaczającego komórkę `A1`. Dzięki temu zyskujemy pewność, że makro będzie działać poprawnie po wprowadzeniu dodatkowych danych.



Uwaga

Kod byłby bardziej uniwersalny, gdyby w zbiorze `PivotFields` zastosowano indeksy zamiast literałów. W ten sposób makro działałoby nawet wtedy, gdy użytkownik zmieniłby nagłówki kolumn. Na przykład bardziej uniwersalny kod zawierałby odwołanie `PivotFields(1)` zamiast `PivotFields("Region")`. Kod zastosowany w pokazanym przykładzie najlepiej pasuje do sytuacji, kiedy układ kolumn nie będzie się zmieniać.

Najlepszym sposobem poznania potrzebnych obiektów, metod i właściwości jest jak zwykle zarejestrowanie wykonywanych działań w makrze. Następnie warto przestudiować tematy pomocy online, aby zrozumieć sposób użycia poszczególnych elementów. Niemal w każdym przypadku zarejestrowane makra wymagają modyfikacji. Można też napisać kod od początku, bez wykorzystywania rejestratora makr, ale to wymaga pewnego doświadczenia.

Złożone tabele przestawne

W tym podrozdziale zaprezentuję kod VBA tworzący stosunkowo złożoną tabelę przestawną.

Dane dla złożonej tabeli przestawnej

Na rysunku 17.3 zaprezentowano fragment bazy danych w arkuszu, zbudowanej z 15 840 wierszy zawierających hierarchiczne dane budżetowe firmy. Korporacja składa się z 5 oddziałów, a każdy oddział z 11 wydziałów. W każdym wydziale są cztery kategorie budżetowe, a każda kategoria zawiera po kilka pozycji. Kwoty budżetu oraz rzeczywiście wydane środki są zapisane dla każdego z 12 miesięcy.

Rysunek 17.3.
Dane z tego skoroszytu będą zestawione w tabeli przestawnej

	A	B	C	D	E	F	G
	ODDZIAŁ	WYDZIAŁ	KATEGORIA	POZYCJA	MIESIĄC	BUDŻET	RZECZYWISTE WYDA
1	Ameryka Pln	Informatyka	Rekompensaty	Wynagrodzenia	Sty	2593	31E
2	Ameryka Pln	Informatyka	Rekompensaty	Zyski	Sty	4496	29E
3	Ameryka Pln	Informatyka	Rekompensaty	Premie	Sty	3768	30E
4	Ameryka Pln	Informatyka	Rekompensaty	Zlecenia	Sty	3133	28E
5	Ameryka Pln	Informatyka	Rekompensaty	Podatki od wynagrodz	Sty	3559	37E
6	Ameryka Pln	Informatyka	Rekompensaty	Szkolenie	Sty	3099	35E
7	Ameryka Pln	Informatyka	Rekompensaty	Konferencje	Sty	2931	31E
8	Ameryka Pln	Informatyka	Rekompensaty	Rozrywka	Sty	2632	26E
9	Ameryka Pln	Informatyka	Rekompensaty	Wynajem	Sty	2633	26E
10	Ameryka Pln	Informatyka	Rekompensaty	Dzierżawa	Sty	3450	26E
11	Ameryka Pln	Informatyka	Rekompensaty	Narzędzia	Sty	4111	30E
12	Ameryka Pln	Informatyka	Rekompensaty	Remonty	Sty	3070	28E
13	Ameryka Pln	Informatyka	Rekompensaty	Rachunki telefoniczne	Sty	3827	43E
14	Ameryka Pln	Informatyka	Rekompensaty	Inne	Sty	3843	33E
15	Ameryka Pln	Informatyka	Rekompensaty	Materiały i serwis	Sty	2642	32E
16	Ameryka Pln	Informatyka	Rekompensaty	Materiały i serwis	Sty	3052	40E
17	Ameryka Pln	Informatyka	Rekompensaty	Materiały i serwis	Sty	4346	33E
18	Ameryka Pln	Informatyka	Rekompensaty	Materiały i serwis	Sty	2669	37E
19	Ameryka Pln	Informatyka	Rekompensaty	Materiały i serwis	Sty	3328	31E
20	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	3088	27E
21	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	4226	26E
22	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	3780	35E
23	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	3893	36E
24	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	2651	43E
25	Ameryka Pln	Informatyka	Rekompensaty	Sprzęt	Sty	3604	35E
26	Ameryka Pln	Kadry	Rekompensaty	Wynagrodzenia	Sty	2659	44E
27	Ameryka Pln	Kadry	Rekompensaty	Zyski	Sty	3020	26E
28	Ameryka Pln	Kadry	Rekompensaty	Premie	Sty		



Ten przykład jest dostępny na płycie CD-ROM dołączonej do książki.

Na rysunku 17.4. pokazano tabelę przestawną utworzoną na podstawie zaprezentowanych wyżej danych. Warto zwrócić uwagę, że tabela przestawna zawiera obliczane pole Odchylenie oraz cztery obliczane pola Kw1, Kw2, Kw3 i Kw4, zawierające podsumowania dla poszczególnych kwartałów.

Kod tworzący tabelę przestawną

Kod VBA z listingu 17.2 tworzy tabelę przestawną.

Listing 17.2. Tworzenie tabeli przestawnej dla wielu oddziałów

```
Sub CreatePivotTable()
    Dim PTCache As PivotCache
    Dim PT As PivotTable

    Application.ScreenUpdating = False
```

Rysunek 17.4.
Tabela przestawna
utworzona
na podstawie danych
z rysunku 17.3

WYDZIAŁ	Dane	MIESIAC	Sty	Lut	Mar	KW1	Kwi	Maj	Cze
Bezpieczeństwo	Budżet (zł)		419195	419294	413258	1251747	421700	421875	
	Rzeczywiste wydatki(zł)		409486	418697	427401	1255584	419221	421266	
	Odchylenie (zł)		9709	597	-14143	-3837	2479	609	
Dostawy	Budżet (zł)		429595	429917	423975	1263467	419617	433168	
	Rzeczywiste wydatki(zł)		413635	413777	414932	1242544	411949	410561	
	Odchylenie (zł)		15760	16140	9043	40943	7668	22607	
Informatyka	Budżet (zł)		422197	422057	419659	1263913	417260	422848	
	Rzeczywiste wydatki(zł)		414743	438990	430545	1284278	424214	411775	
	Odchylenie (zł)		7454	-16933	-10886	-20365	-6954	11073	
Kadry	Budżet (zł)		422053	425313	418634	1266000	423038	423514	
	Rzeczywiste wydatki(zł)		424934	429275	407053	1261262	429187	410258	
	Odchylenie (zł)		-2881	-3962	11581	4738	-6149	13256	
Księgowość	Budżet (zł)		422455	433317	420522	1276294	417964	411820	
	Rzeczywiste wydatki(zł)		422662	413163	416522	1252347	420672	431303	
	Odchylenie (zł)		-207	20154	4000	23947	-2708	-19483	
Operacje	Budżet (zł)		413530	427975	419527	1261032	422299	415296	
	Rzeczywiste wydatki(zł)		415819	406592	426627	1249238	418223	431307	
	Odchylenie (zł)		-2289	21383	-7300	11794	4076	-16009	
Public Relations	Budżet (zł)		424896	414507	415179	1254582	417100	426223	
	Rzeczywiste wydatki(zł)		413526	414084	415476	1243086	414040	396652	
	Odchylenie (zł)		11370	423	-297	11496	3060	29571	
Reklama	Budżet (zł)		424590	419331	417949	1261870	420324	427150	

- Usunięcie arkusza ArkuszTabeliPrzestawnej, jeśli istnieje
 - On Error Resume Next
 - Application.DisplayAlerts = False
 - Sheets("ArkuszTabeliPrzestawnej").Delete
 - On Error GoTo 0
 - Application.DisplayAlerts = True
- Utworzenie bufora tabeli przestawnej
 - Set PTCache = ActiveWorkbook.PivotCaches.Add(_
 - SourceType:=xlDatabase, _
 - SourceData:="Range("A1").CurrentRegion.Address)
- Dodanie nowego arkusza
 - Worksheets.Add
 - ActiveSheet.Name = "ArkuszTabeliPrzestawnej"
- Utworzenie tabeli przestawnej na podstawie bufora
 - Set PT = PTCache.CreatePivotTable(_
 - TableDestination:=Sheets("ArkuszTabeliPrzestawnej").Range("A1"), _
 - TableName:="TabelaPrzestawnaBudżetu")
- With PT
 - Dodanie pól
 - .PivotFields("WYDZIAŁ").Orientation = xlRowField
 - .PivotFields("MIESIAC").Orientation = xlColumnField
 - .PivotFields("ODDZIAŁ").Orientation = xlPageField
 - .PivotFields("BUDŻET").Orientation = xlDataField
 - .PivotFields("RZECZYWISTE WYDATKI").Orientation = xlDataField
 - Dodanie pola służącego do obliczania odchylenia
 - .CalculatedFields.Add "Odchylenie", "=BUDŻET-RZECZYWISTE WYDATKI"
 - .PivotFields("Odchylenie").Orientation = xlDataField

```
' Dodanie pól obliczanych
.PivotFields("MIESIĄC").CalculatedItems.Add _
    "Kw1", "= Sty+Lut+Mar"
.PivotFields("MIESIĄC").CalculatedItems.Add _
    "Kw2", "= Kwi+Maj+Cze"
.PivotFields("MIESIĄC").CalculatedItems.Add _
    "Kw3", "= Lip+Sie+Wrz"
.PivotFields("MIESIĄC").CalculatedItems.Add _
    "Kw4", "= Paź+Lis+Gru"

' Przeniesienie obliczanych pozycji
.PivotFields("MIESIĄC").PivotItems("Kw1").Position = 4
.PivotFields("MIESIĄC").PivotItems("Kw2").Position = 8
.PivotFields("MIESIĄC").PivotItems("Kw3").Position = 12
.PivotFields("MIESIĄC").PivotItems("Kw4").Position = 16

' Modyfikacja podpisów
.PivotFields("Suma z BUDŻET").Caption = "Budżet (zł)"
.PivotFields("Suma z RZECZYWISTE WYDATKI").Caption = "Rzeczywiste wydatki(zł)"
.PivotFields("Suma z Odchylenie").Caption = "Odchylenie (zł)"
End With

Application.ScreenUpdating = True
End Sub
```

Jak działa złożona tabela przestawna?

Procedura `CreatePivotTable` z listingu 17.2 najpierw usuwa arkusz `ArkuszTabeliPrzestawnej`, jeżeli taki wcześniej istniał. Następnie tworzy obiekt `PivotCache`, wstawia nowy arkusz o nazwie `ArkuszTabeliPrzestawnej` i w końcu tworzy tabelę przestawną. Do tabeli przestawnej dodawane są następujące pola:

- Wydział — pole wiersza;
- Miesiąc — pole kolumny;
- Oddział — pole strony;
- Budżet — pole danych;
- Rzeczywiste wydatki — pole danych.

Następnie wykorzystano metodę `Add` zbioru `CalculatedFields` w celu utworzenia obliczanego pola `Odchylenie`, którego wartość jest wyliczana poprzez odjęcie wartości faktycznie wydanej kwoty (pole `RZECZYWISTE WYDATKI`) od kwoty budżetu (pole `BUDŻET`). Cztery obliczane pola służą do wyliczenia kwartalnych podsumowań. Domyślnie obliczane pola są dodawane po prawej stronie tabeli przestawnej, a zatem potrzebny jest dodatkowy kod, aby umieścić je obok miesięcy, których dotyczą (np. pole `Kw1` jest umieszczone za polem `Mar`). Na koniec kod modyfikuje podpisy wyświetlane w tabeli przestawnej (np. nazwa `Suma z BUDŻET` jest zastępowana nazwą `Budżet (zł)`).



Tworząc tę procedurę, najpierw wygenerowałem kod za pomocą rejestratora makr, a następnie go poprawiłem, aby stał się czytelniejszy i wydajniejszy.

Tworzenie tabel przestawnych na podstawie zewnętrznej bazy danych

W poprzednim przykładzie źródłem danych był arkusz. Jak wiadomo, Excel umożliwia wykorzystanie zewnętrznych źródeł danych do tworzenia tabel przestawnych. Przykład zaprezentowany w tym podrozdziale demonstruje kod VBA potrzebny do utworzenia tabeli przestawnej na podstawie danych zapisanych w bazie danych *Access*.



Baza danych *Accessa* zawiera pojedynczą tabelę, w której umieszczono te same dane, których użyto w poprzednim przykładzie.

Kod z listingu 17.3 tworzy tabelę przestawną. Założono, że plik bazy danych *budzet.mdb* jest zapisany w tym samym katalogu, co arkusz.

Listing 17.3. *Generowanie tabeli przestawnej na podstawie danych z zewnętrznej bazy danych*

```
Sub CreatePivotTableFromDB()
    Dim PTCache As PivotCache
    Dim PT As PivotTable
    Dim DBFile As String
    Dim ConString As String
    Dim QueryString As String

    ' Usunięcie arkusza ArkPrzest, jeżeli istnieje
    On Error Resume Next
    Application.DisplayAlerts = False
    Sheets("ArkPrzest").Delete
    On Error GoTo 0

    ' Utworzenie bufora tabeli przestawnej
    Set PTCache = ActiveWorkbook.PivotCaches.Add _
        (SourceType:=xlExternal)

    ' Ścieżka do pliku bazy danych
    DBFile = ThisWorkbook.Path & "\budzet.mdb"

    ' Łańcuch połączenia
    ConString = "ODBC:DSN=MS Access Database;DBQ=" & DBFile

    ' Zapytanie
    QueryString = "SELECT * FROM BUDŻET"

    With PTCache
        .Connection = ConString
        .CommandText = QueryString
    End With

    ' Dodanie arkusza
    Worksheets.Add
    ActiveSheet.Name = "ArkPrzest"

    ' Utworzenie tabeli przestawnej
    Set PT = PTCache.CreatePivotTable( _
```

```

TableDestination:=Sheets("ArkPrzest").Range("A1"), _
TableName:="BudzetPrzest")

With PT
    Dodanie pól
    .PivotFields("WYDZIAŁ").Orientation = xlRowField
    .PivotFields("MIESIĄC").Orientation = xlColumnField
    .PivotFields("ODDZIAŁ").Orientation = xlPageField
    .PivotFields("BUDŻET").Orientation = xlDataField
    .PivotFields("RZECZYWISTE WYDATKI").Orientation = xlDataField
End With
End Sub

```

Argument `SourceType` metody `Add` zbioru `PivotCaches` określono jako `xlExternal`. W przykładzie zaprezentowanym w poprzednim podrozdziale (w którym wykorzystano dane z bazy danych w arkuszu) argument `SourceType` był typu `xlDatabase`. Obiekt `PivotCache` w celu pobrania danych z pliku zewnętrznego potrzebuje następujących informacji:

- *łańcucha połączenia* opisującego typ źródła danych oraz nazwę pliku,
- *łańcucha zapytania*, którym jest instrukcja SQL określająca zwracane rekordy i pola.

W tym przykładzie łańcuch połączenia określa źródło danych ODBC, którym jest plik *Accessa* o nazwie *budzet.mdb*. Łańcuch zapytania określa w tym przypadku całą tabelę `BUDŻET`. Informacje te są przekazywane do obiektu `PivotCache` poprzez ustawienie właściwości `Connection` oraz `CommandText`. Po zapisaniu danych w buforze tabeli przestawnej następuje utworzenie tabeli za pomocą metody `CreatePivotTable`.



Uwaga

SQL jest standardowym językiem służącym do tworzenia zapytań do bazy danych. Więcej informacji można uzyskać w pomocy online. Warto również zakupić książkę poświęconą temu językowi, co pozwoli poznać go dokładniej.

Jednoczesne tworzenie wielu tabel przestawnych

W ostatnim przykładzie utworzymy kilka tabel przestawnych, które będą zawierały podsumowania danych pobranych z ankiety wypełnianej przez klientów. Dane są zapisane w bazie danych w arkuszu (rysunek 17.5) złożonej ze 100 wierszy. Każdy wiersz zawiera dane o płci respondenta oraz odpowiedź od 1 do 5 dla każdej z 14 pozycji ankiety.

Na rysunku 17.6 pokazano kilka tabel przestawnych. Dla każdej pozycji ankiety utworzono dwie tabele przestawne — jedna wyświetla dane w procentach, druga rzeczywiste wartości.

Kod VBA z listingu 17.4 tworzy te tabele.

Listing 17.4. Tworzenie wielu tabel przestawnych na podstawie danych ze złożonej zewnętrznej bazy danych

```

Sub MakePivotTables()
    ' Ta procedura tworzy 28 tabel przestawnych
    Dim PTCache As PivotCache
    Dim PT As PivotTable

```

Rysunek 17.5.
Wyniki ankiety będą podsumowane w kilku tabelach przestawnych

	A	B	C	D	E	F	G	H	I	J	
	Nazwisko	Płeć	Lokalizacje magazynów są dogodne	Godziny otwarcia są dogodne	Magazyny są dobrze utrzymane	Dobry kontakt telefoniczny	Podoba mi się wirtyna WWW	Pracownicy są mili	Pracownicy oferują pomoc	Pracownicy mają wiedzę	Ko ji
1											
2	Osoba1	Mężczyzna	1	4	4	4	1	1	2	1	
3	Osoba2	Kobieta	2	5	1	1	4	2	4	3	
4	Osoba3	Mężczyzna	1	1	4	2	3	3	2	1	
5	Osoba4	Mężczyzna	2	1	3	5	1	2	3	4	
6	Osoba5	Kobieta	2	2	5	5	4	2	1	5	
7	Osoba6	Kobieta	2	4	3	3	1	1	4	4	
8	Osoba7	Kobieta	2	4	5	4	5	3	2	5	
9	Osoba8	Mężczyzna	3	2	1	2	3	4	3	1	
10	Osoba9	Kobieta	3	4	4	4	5	1	4	1	
11	Osoba10	Mężczyzna	2	1	5	5	5	1	4	1	
12	Osoba11	Mężczyzna	4	3	3	2	1	2	4	2	
13	Osoba12	Kobieta	2	1	4	5	5	5	3	1	
14	Osoba13	Kobieta	4	3	4	3	2	5	3	3	
15	Osoba14	Kobieta	2	3	4	2	1	1	4	2	
16	Osoba15	Kobieta	1	3	5	1	2	2	4	1	
17	Osoba16	Mężczyzna	1	4	1	3	4	3	4	4	
18	Osoba17	Kobieta	3	4	3	5	5	4	4	3	
19	Osoba18	Mężczyzna	1	5	5	3	5	3	4	2	
20	Osoba19	Kobieta	1	3	5	4	5	5	5	1	
21	Osoba20	Mężczyzna	2	2	5	2	2	5	5	3	
22	Osoba21	Mężczyzna	3	4	1	4	5	1	3	1	
23	Osoba22	Mężczyzna	2	1	5	5	5	1	2	1	
24	Osoba23	Mężczyzna	4	3	4	2	1	2	1	2	
25	Osoba24	Kobieta	1	1	2	5	5	5	3	1	

	A	B	C	D	E	F	G	H	I
	Procent	Płeć				Liczba odpowiedzi	Płeć		
1	Procent								
2	Lokalizacje magazynów są dogodne	Kobieta	Mężczyzna	Suma końcowa	Lokalizacje magazynów są dogodne		Kobieta	Mężczyzna	Suma końcowa
3	Kategorycznie się nie zgadzam	42,56%	49,06%	46,00%	Kategorycznie się nie zgadzam		20	26	46
4	Nie zgadzam się	27,66%	20,75%	24,00%	Nie zgadzam się		13	11	24
5	Jestem niezdecydowany	21,28%	11,32%	16,00%	Jestem niezdecydowany		10	6	16
6	Zgadzam się	6,38%	18,87%	13,00%	Zgadzam się		3	10	13
7	Całkowicie się zgadzam	2,13%	0,00%	1,00%	Całkowicie się zgadzam		1		1
8	Suma końcowa	100,00%	100,00%	100,00%	Suma końcowa		47	53	100
9									
10									
11									
12	Procent								
13	Godziny otwarcia są dogodne	Kobieta	Mężczyzna	Suma końcowa	Godziny otwarcia są dogodne		Kobieta	Mężczyzna	Suma końcowa
14	Kategorycznie się nie zgadzam	12,77%	16,98%	15,00%	Kategorycznie się nie zgadzam		6	9	15
15	Nie zgadzam się	10,64%	15,09%	13,00%	Nie zgadzam się		5	8	13
16	Jestem niezdecydowany	40,43%	30,19%	35,00%	Jestem niezdecydowany		19	16	35
17	Zgadzam się	31,91%	28,30%	30,00%	Zgadzam się		15	15	30
18	Całkowicie się zgadzam	4,26%	9,43%	7,00%	Całkowicie się zgadzam		2	5	7
19	Suma końcowa	100,00%	100,00%	100,00%	Suma końcowa		47	53	100
20									
21									
22									
23	Procent								
24	Magazyny są dobrze utrzymane	Kobieta	Mężczyzna	Suma końcowa	Magazyny są dobrze utrzymane		Kobieta	Mężczyzna	Suma końcowa
25	Kategorycznie się nie zgadzam	12,77%	15,09%	14,00%	Kategorycznie się nie zgadzam		6	8	14
26	Nie zgadzam się	8,51%	5,66%	7,00%	Nie zgadzam się		4	3	7
27	Jestem niezdecydowany	27,66%	20,75%	24,00%	Jestem niezdecydowany		13	11	24
28	Zgadzam się	36,17%	37,74%	37,00%	Zgadzam się		17	20	37

Rysunek 17.6. Tabele przestawne utworzone za pomocą procedury VBA

```

Dim SummarySheet As Worksheet
Dim ItemName As String
Dim Row As Integer, i As Integer

Application.ScreenUpdating = False

' Usunięcie arkusza "Podsumowanie", jeżeli istnieje
On Error Resume Next
Application.DisplayAlerts = False
Sheets("Podsumowanie").Delete

On Error GoTo 0

' Dodanie arkusza "Podsumowanie"
Set SummarySheet = Worksheets.Add
ActiveSheet.Name = "Podsumowanie"

' Utworzenie bufora tabeli przestawnej
Set PTCache = ActiveWorkbook.PivotCaches.Add( _
    SourceType:=xlDatabase, _
    SourceData:=Sheets("DaneZAnkiety").Range("A1"). _
    CurrentRegion.Address)

Row = 1
For i = 1 To 14
    ItemName = Sheets("DaneZAnkiety").Cells(1, i + 2)

    ' Utworzenie tabeli przestawnej
    Set PT = PTCache.CreatePivotTable _
        (TableDestination:=SummarySheet.Cells(Row, 6), _
        TableName:=ItemName)
    Row = Row + 11

    ' Dodanie pól
    With PT.PivotFields(ItemName)
        .Orientation = xlDataField
        .Name = "Liczba odpowiedzi"
        .Function = xlCount
    End With

    PT.AddFields RowFields:=Array(ItemName, "Dane")
    PT.PivotFields("Płeć").Orientation = xlColumnField
Next i

Row = 1
For i = 1 To 14
    ItemName = Sheets("DaneZAnkiety").Cells(1, i + 2)

    ' Utworzenie tabeli przestawnej
    Set PT = PTCache.CreatePivotTable _
        (TableDestination:=SummarySheet.Cells(Row, 1), _
        TableName:=ItemName & "2")
    Row = Row + 11

    With PT.PivotFields(ItemName)
        .Orientation = xlDataField
        .Name = "Procent"
        .Function = xlCount
    End With

```

```

        .Calculation = xlPercentOfColumn
    End With
    PT.AddFields RowFields:=Array(ItemName, "Dane")
    PT.PivotFields("Płeć").Orientation = xlColumnField
Next i

' Zamiana liczb na opisowy tekst
Summary Sheet.Activate
With Range("A:A,F:F")
    .Replace "1", "Kategorycznie się nie zgadzam"
    .Replace "2", "Nie zgadzam się"
    .Replace "3", "Jestem niezdecydowany"
    .Replace "4", "Zgadzam się"
    .Replace "5", "Całkowicie się zgadzam"
End With

' Skorygowanie szerokości kolumn
Columns("A:I").EntireColumn.AutoFit
End Sub

```

Tabele przestawne są tworzone w dwóch pętlach, wszystkie na podstawie jednego obiektu PivotCache. Początek każdej kolejnej tabeli przestawnej jest wyliczany na podstawie zmiennej Row. Po utworzeniu tabeli kod zamienia kategorie liczbowe w pierwszej kolumnie na tekst (np. wartość 1 jest zastępowana tekstem *Kategorycznie się nie zgadzam*). Na końcu są korygowane szerokości kolumn.

Modyfikowanie tabel przestawnych

Tabele przestawne Excela są elastyczne. Użytkownicy mogą z łatwością zamienić pole wiersza z polem kolumny lub ukryć niektóre pozycje w tabeli, nieistotne w danym momencie. Można wprowadzić własne elementy interfejsu, które jeszcze bardziej ułatwią modyfikacje. W przykładzie zaprezentowanym w tym podrozdziale pokazano tabelę przestawną, której wyglądem można sterować za pomocą przełączników oraz pól wyboru tak, jak pokazano na rysunku 17.7.

Rysunek 17.7.
Użytkownik może skorzystać z formantów w celu modyfikowania tabel przestawnych

RepHandl	Kw1	Kw2	Kw3	Kw4	Suma końcowa
Andrzej	41 817	41 236	45 122	45 691	173 866
Bogdan	38 700	42 691	40 357	39 767	161 715
Czesław	35 204	36 303	35 689	30 040	137 436
Daniel	51 268	51 659	53 611	51 736	208 464

Tabela przestawna zawiera cztery dodatkowe pola obliczane (Kw1, Kw2, Kw3 i Kw4), które służą do obliczania podsumowań kwartalnych. Kod wykonywany w przypadku zaznaczenia pierwszej wartości przełącznika (*Tylko miesiące*) pokazuje listing 17.5. Procedura jest prosta, przypomina procedury obsługi zdarzeń dla innych przełączników.

Listing 17.5. Modyfikacja tabeli przestawnej

```
Private Sub OptionButton1_Click()  
    ' Tylko miesiące  
    Application.ScreenUpdating = False  
    With ActiveSheet.PivotTables(1).PivotFields("Miesiąc")  
        .PivotItems("Sty").Visible=True  
        .PivotItems("Lut").Visible=True  
        .PivotItems("Mar").Visible=True  
        .PivotItems("Kwi").Visible=True  
        .PivotItems("Maj").Visible=True  
        .PivotItems("Cze").Visible=True  
        .PivotItems("Lip").Visible=True  
        .PivotItems("Sie").Visible=True  
        .PivotItems("Wrz").Visible=True  
        .PivotItems("Paź").Visible=True  
        .PivotItems("Lis").Visible=True  
        .PivotItems("Gru").Visible=True  
        .PivotItems("Kw1").Visible=False  
        .PivotItems("Kw2").Visible=False  
        .PivotItems("Kw3").Visible=False  
        .PivotItems("Kw4").Visible=False  
    End With  
End Sub
```

Pola wyboru powodują włączanie (wyłączenie) podsumowań dla wierszy i kolumn. Procedury obsługi tych zdarzeń zaprezentowano poniżej:

```
Private Sub CheckBox1_Click()  
    ' Podsumowania kolumn  
    Application.ScreenUpdating = False  
    ActiveSheet.PivotTables(1).ColumnGrand = CheckBox1.Value  
End Sub  
  
Private Sub CheckBox2_Click()  
    ' Podsumowania wierszy  
    Application.ScreenUpdating = False  
    ActiveSheet.PivotTables(1).RowGrand = CheckBox2.Value  
End Sub
```

Tabele przestawne można oczywiście modyfikować w różny sposób. Jak wspomniałem, najprostszym sposobem utworzenia kodu VBA, który modyfikuje tabele przestawne, jest wykorzystanie rejestratora makr podczas ręcznych modyfikacji. Następnie wystarczy wprowadzić poprawki w kodzie i skopiować go do procedur obsługi zdarzeń wybranych formantów.