

Richard Mansfield

VBA

dla Microsoft® Office 365®
i Office 2019®

Helion 

 **SYBEX**
A Wiley Brand

Tytuł oryginału: Mastering VBA for Microsoft Office 365

Tłumaczenie: Radosław Meryk

ISBN: 978-83-283-6513-1

Copyright © 2019 by John Wiley & Sons, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license with the original publisher John Wiley & Sons, Inc.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise without either the prior written permission of the Publisher

The SYBEX Brand trade dress is a trademark of John Wiley & Sons, Inc. in the United States and/or other countries. Used by permission.

Translation copyright © 2020 by Helion SA

Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Microsoft and Office 365 are registered trademarks of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Szata graficzna SYBEX jest znakiem towarowym firmy John Wiley & Sons, Inc. w Stanach Zjednoczonych i/lub innych krajach. Wykorzystano za pozwoleniem.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/vbaoff>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wprowadzenie	25
Część I • Rejestrowanie makr i wprowadzenie do VBA	33
Rozdział 1. Rejestrowanie i uruchamianie makr w aplikacjach pakietu Office	35
Czym jest VBA i co można zrobić za jego pomocą?	35
Różnica między językami Visual Basic a Visual Basic for Applications	36
Co to jest Visual Basic .NET i Visual Basic Express?	37
Podstawowe wiadomości o makrach	38
Rejestrowanie makr	39
Wyświetlanie zakładki Deweloper na Wstążce	39
Planowanie makra	40
Uruchamianie Rejestratora makr	42
Nazewnictwo makr	45
Wybór sposobu uruchomienia nowego makra	49
Uruchamianie makra	55
Rejestrowanie przykładowego makra w programie Word	57
Rejestrowanie przykładowego makra w Excelu	59
Tworzenie skróty makr osobistych, jeśli nie stworzono go wcześniej	59
Rejestrowanie makra	60
Określanie sposobu uruchamiania istniejącego makra	61
Przypisywanie makra do przycisku paska narzędzi Szybki dostęp w programie Word	62
Przypisywanie makra do klawisza skrótu	62
Usuwanie makra	62
Pytania kontrolne i zadania	64
Rozdział 2. Wprowadzenie do programu Visual Basic Editor	65
Otwieranie edytora VBE	65
Otwieranie VBE z wybranym makrem	66
Otwieranie edytora VBE bezpośrednio	67
Przejsięcie do makra	67
Korzystanie z głównych okien edytora VBE	68
Project Explorer	68
Przeglądarka obiektów	72
Okno kodu	73
Okno Properties	77
Okno Immediate	79
Ustawianie właściwości projektu	80
Personalizacja programu Visual Basic Editor	82
Wybór ustawień edytora i preferencji wyświetlania	83
Wybór i rozmieszczanie okien edytora	90

Dostosowywanie paska narzędzi i paska menu	90
Dostosowywanie przybornika	91
Pytania kontrolne i zadania	95

Rozdział 3. Edycja zarejestrowanych makr 97

Testowanie makra w programie Visual Basic Editor	98
Uruchamianie makra krok po kroku	99
Ustawianie pułapek	101
Komentowanie wierszy kodu	102
Wyjście z krokowego wykonywania makra	103
Edycja makra programu Word	103
Uruchamianie makra Transpose_Word_Right krok po kroku	104
Uruchamianie makra Transpose_Word_Right	105
Tworzenie makra Transpose_Word_Left	105
Zapisywanie pracy	106
Edycja makra Excela	107
Odkrywanie skoroszytu makr osobistych	107
Otwarcie makra do edycji	108
Edycja makra	109
Edycja makra PowerPoint	111
Zmniejszenie rozmiaru makra	114
Zapisywanie pracy	115
Pytania kontrolne i zadania	117

Rozdział 4. Tworzenie kodu w edytorze VBE od podstaw 119

Konfigurowanie edytora VBE do tworzenia makr	119
Tworzenie procedur dla Worda	121
Tworzenie makra dla programu Excel	126
Tworzenie procedury dla PowerPointa	130
Tworzenie procedury dla Accessa	136
Pytania kontrolne i zadania	137

Część II • Korzystanie z VBA 139

Rozdział 5. Podstawy składni VBA 141

Przygotuj się	141
Procedury	142
Funkcje	143
Procedury Sub	143
Instrukcje	144
Słowa kluczowe	147
Wyrażenia	148
Operatory	148
Zmienne	148
Stałe	150
Argumenty	150
Określanie nazw argumentów, a ich pomijanie	151
Kiedy stosować nawiasy wokół listy argumentów	152

Obiekty	152
Kolekcje	153
Właściwości	153
Metody	153
Zdarzenia	154
Pytania kontrolne i zadania	156
Rozdział 6. Zmienne, stałe i wyliczenia	157
Korzystanie ze zmiennych	158
Wybór nazw dla zmiennych	158
Deklarowanie zmiennych	160
Wybór zakresu i czasu życia zmiennej	163
Określanie typu danych dla zmiennej	169
Korzystanie ze stałych	175
Deklarowanie własnych stałych	175
Wybór zasięgu lub czasu życia dla stałych	176
Wyliczenia	176
Pytania kontrolne i zadania	178
Rozdział 7. Zmienne tablicowe	179
Czym są tablice?	179
Deklarowanie tablicy	181
Przechowywanie wartości w tablicy	183
Tablice wielowymiarowe	183
Deklarowanie tablic dynamicznych	184
Zmiana wymiarów tablicy	185
Zwracanie informacji z tablicy	185
Usuwanie tablicy	186
Ustalanie, czy zmienna jest tablicą	186
Wyznaczanie granic tablic	186
Sortowanie tablic	186
Przeszukiwanie tablicy	190
Liniowe przeszukiwanie tablicy	191
Binarne przeszukiwanie tablicy	195
Pytania kontrolne i zadania	200
Rozdział 8. Wyszukiwanie obiektów, metod i właściwości, których potrzebujesz	203
Czym jest obiekt?	203
Zalety programowania obiektowego	203
Obiekty „kreowane”	205
Właściwości	205
Metody	206
Kolekcje	208
Korzystanie z obiektu w kolekcji	209
Dodawanie obiektu do kolekcji	209
Znajdowanie potrzebnych obiektów	209
Korzystanie z Rejestratora makr, aby dodać kod do potrzebnych obiektów	210
Korzystanie z Przeglądarki obiektów	212

Korzystanie z pomocy, aby znaleźć potrzebny obiekt	218
Korzystanie z własności Auto List Members	221
Używanie zmiennych typu Object do reprezentowania obiektów	222
Programowanie zespołowe i OOP	225
Pytania kontrolne i zadania	227

Część III • Podejmowanie decyzji z wykorzystaniem pętli i funkcji229

Rozdział 9. Korzystanie z funkcji wbudowanych	231
Czym są funkcje?	231
Korzystanie z funkcji	233
Przekazywanie argumentów do funkcji	235
Korzystanie z funkcji do konwersji danych	236
Funkcja ASC do wyznaczania kodu znaku	237
Funkcja Val do wyodrębnienia liczby z początku ciągu znaków	238
Korzystanie z funkcji Str w celu konwersji liczby na łańcuch znaków	239
Korzystanie z funkcji Format do formatowania wyrażeń	240
Korzystanie z predefiniowanych formatów liczbowych	241
Tworzenie formatu liczbowego	241
Tworzenie formatu dat i godzin	242
Tworzenie formatu ciągów znaków	244
Wykorzystanie funkcji Chr i stałych do wprowadzania znaków specjalnych do ciągu	244
Korzystanie z funkcji do wykonywania operacji na zmiennych	245
Korzystanie z funkcji Left, Right i Mid w celu uzyskania fragmentu ciągu znaków	247
Korzystanie z funkcji Left	247
Korzystanie z funkcji Right	248
Korzystanie z funkcji Mid	248
Wykorzystanie funkcji InStr i InStrRev w celu znalezienia ciągu znaków w innym ciągu znaków	250
Korzystanie z funkcji LTrim, RTrim i Trim do usuwania spacji z ciągu znaków	252
Wykorzystanie funkcji Len do sprawdzania długości ciągu znaków	253
Korzystanie z funkcji StrConv, LCase i UCase do zmiany wielkości liter w ciągu znaków	255
Korzystanie z funkcji StrComp do porównywania jabłek z jabłkami	256
Korzystanie z funkcji matematycznych języka VBA	257
Korzystanie z funkcji VBA do przetwarzania dat i godzin	258
Korzystanie z funkcji DatePart do parsowania dat	259
Obliczanie przedziałów czasowych za pomocą funkcji DateDiff	260
Korzystanie z funkcji DateAdd do dodawania lub odejmowania czasu od daty	260
Funkcje do zarządzania plikami	261
Sprawdzanie, czy plik istnieje, za pomocą funkcji Dir	261
Zwracanie bieżącej ścieżki	263
Pytania kontrolne i zadania	263
Rozdział 10. Tworzenie własnych funkcji	265
Składowe instrukcji Function	266
Tworzenie funkcji	268
Tworzenie funkcji ręcznie	268
Tworzenie funkcji za pomocą okna dialogowego Add Procedure	268

Przekazywanie argumentów do funkcji	269
Deklarowanie typów danych argumentów	270
Określanie argumentu opcjonalnego	270
Zarządzanie zasięgiem funkcji	271
Przykłady funkcji dla wszystkich aplikacji pakietu Office obsługujących VBA	271
Jak funkcje zwracają informacje?	272
Zwracanie z funkcji danych tekstowych	273
Tworzenie funkcji dla programu Word	276
Tworzenie funkcji dla programu Excel	277
Tworzenie funkcji dla PowerPointa	279
Tworzenie funkcji dla programu Access	280
Pytania kontrolne i zadania	282
Rozdział 11. Podejmowanie decyzji w kodzie	285
Jak w VBA wykonujemy porównania?	286
Testowanie wielu warunków z wykorzystaniem operatorów logicznych	287
Bloki If	290
If...Then	290
Instrukcje If...Then...Else	292
Instrukcje If...Then...ElseIf...Else	294
Tworzenie pętli z instrukcjami If i GoTo	298
Zagnieżdżanie bloków If	299
Bloki Select Case	301
Składnia	302
Przykład	302
Kolejność ma znaczenie	304
Pytania kontrolne i zadania	305
Rozdział 12. Korzystanie z pętli do powtarzania działań	307
Kiedy należy używać pętli?	307
Podstawy działania pętli	308
Korzystanie z pętli For... do wykonywania stałej liczby powtórzeń	308
Pętle For...Next	308
Pętle For Each...Next	316
Wykorzystanie instrukcji Exit For	317
Wykorzystanie pętli Do... w celu wykonania zmiennej liczby powtórzeń	318
Pętle Do While...	318
Pętle Do...Loop While	322
Pętle Do Until...Loop	324
Pętle Do...Loop Until	326
Wykorzystanie instrukcji Exit Do	327
Czy stosowanie instrukcji Exit Do to zła praktyka?	328
Pętle While...Wend	329
Zagnieżdżanie pętli	329
Unikanie pętli nieskończonych	332
Pytania kontrolne i zadania	332

Część IV • Korzystanie z okien MsgBox, InputBox i okien dialogowych335

Rozdział 13. Pobieranie danych od użytkownika za pomocą okien MsgBox i InputBox337

Otwieranie makra	338
Wyświetlanie komunikatów na pasku stanu w Wordzie i Excelu	339
Okna komunikatów	341
Zalety i wady okien komunikatów	341
Składnia użycia okien komunikatów	342
Wyświetlanie prostego okna komunikatu	342
Wyświetlanie wielowierszowego okna komunikatu	343
Wybieranie przycisków dla okna komunikatu	344
Wybór ikony dla okna komunikatu	345
Ustawianie domyślnego przycisku dla okna komunikatu	346
Zarządzanie modalnością okna komunikatu	347
Określanie tytułu okna komunikatu	348
Paski tytułu mogą dostarczać przydatnych informacji	348
Dodawanie przycisku Pomoc w oknach komunikatów	349
Określanie pliku pomocy dla okna komunikatu	350
Stosowanie pewnych argumentów bez innych	350
Pobieranie wartości z okna komunikatu	351
Okna wprowadzania danych	352
Składnia polecenia InputBox	352
Pobieranie danych z okien InputBox	354
Formularze: gdy okna komunikatów i okna wprowadzania danych nie wystarczą	355
Pytania kontrolne i zadania	355

Rozdział 14. Tworzenie prostych, niestandardowych okien dialogowych357

Kiedy należy użyć niestandardowego okna dialogowego?	357
Tworzenie niestandardowych okien dialogowych	358
Projektowanie okien dialogowych	360
Wstawianie formularza użytkownika	360
Wybór ustawień siatki formularza użytkownika	361
Zmiana nazwy formularza użytkownika	363
Dodawanie kontroltek do formularzy użytkownika	365
Grupowanie kontroltek	369
Zmiana nazwy kontroltek	369
Przemieszczanie kontroltek	370
Zmiana podpisu kontrolki	372
Najważniejsze właściwości kontroltek z przybornika	373
Praca z grupami kontroltek	389
Jak grupować kontrolki?	389
Anulowanie pogrupowania kontroltek	390
Ustalanie rozmiaru pogrupowanych kontroltek	390
Usuwanie pogrupowanych kontroltek	390
Praca z jedną kontrolką w grupie	391
Wyrównywanie kontroltek	391
Rozmieszczanie kontroltek	392
Dostosowywanie kolejności przechodzenia klawiszem Tab w formularzu	393

Łączenie formularza z procedurą	394
Ładowanie formularzy i usuwanie ich z pamięci	395
Wyświetlanie i ukrywanie formularza	396
Ustawianie domyślnego przycisku polecenia	396
Odczytywanie informacji wprowadzonych przez użytkownika w oknie dialogowym	397
Zwracanie tekstu wprowadzonego w polu tekstowym	397
Zwracanie wartości przycisku opcji	398
Zwracanie właściwości Value z pola wyboru	398
Zwracanie właściwości Value z pola listy	399
Zwracanie właściwości Value z pola kombi	400
Przykłady powiązań formularzy z procedurami	400
Przykład dla Worda: procedura przenoszenia akapitu	400
Przykład ogólny: otwieranie pliku z pola listy	410
Budowanie formularza użytkownika	411
Tworzenie kodu obsługi formularza użytkownika	412
Korzystanie z wbudowanych okien dialogowych aplikacji	415
Wyświetlanie wbudowanego okna dialogowego	416
Ustawianie i przywracanie opcji we wbudowanych oknach dialogowych	419
Który przycisk wybrał użytkownik w oknie dialogowym?	420
Określanie limitu czasu dla okna dialogowego	420
Pytania kontrolne i zadania	421
Rozdział 15. Tworzenie złożonych formularzy	423
Tworzenie złożonych okien dialogowych i korzystanie z nich	424
Aktualizowanie okna dialogowego w celu odzwierciedlenia ustawień wprowadzonych przez użytkownika	424
Odsłanianie ukrytej części formularza	424
Śledzenie procedury w formularzu	429
Korzystanie z wielostronicowych okien dialogowych oraz kontroltek TabStrip	431
Tworzenie niemodalnych okien dialogowych	441
Określanie lokalizacji formularza na ekranie	442
Wykorzystanie zdarzeń do zarządzania formularzami	443
Zdarzenia unikatowe dla obiektu UserForm	446
Zdarzenia, które dotyczą zarówno formularzy użytkownika, jak i kontroltek reprezentujących kontenery	450
Zdarzenia, które dotyczą wielu lub większości kontroltek	455
Zdarzenia, które dotyczą tylko kilku kontroltek	466
Zadania i pytania kontrolne	467

Część V • Tworzenie skutecznego kodu

Rozdział 16. Budowanie modułowego kodu i korzystanie z klas	471
Tworzenie kodu modułowego	471
Czym jest modułowy kod?	472
Zalety używania modułowego kodu	472
Jak tworzyć modułowy kod?	473
Organizowanie kodu w modułach	473
Wywoływanie procedur	473

Wprowadzanie w kodzie ulepszeń logicznych	475
Wizualne udoskonalenia kodu	481
Tworzenie klas i korzystanie z nich	486
Co można zrobić z modułami klas?	487
Krótki przegląd klas	487
Planowanie klasy	488
Tworzenie modułu klasy	488
Nadawanie klasie nazwy	489
Ustawianie właściwości Instancing	489
Deklarowanie zmiennych i stałych klasy	489
Dodawanie właściwości do klasy	490
Dodawanie metod do klasy	494
Korzystanie z klasy	495
Pytania kontrolne i zadania	497
Rozdział 17. Debugowanie kodu i obsługa błędów	499
Zasady debugowania	499
Różne rodzaje błędów	501
Błędy językowe	501
Błędy kompilacji	501
Błędy fazy wykonania	504
Błędy logiki programu	505
Narzędzia debugowania w VBA	506
Tryb pułapki	507
Polecenia Step Over i Step Out	508
Okno Locals	509
Okno Watch	510
Okno Immediate	513
Okno dialogowe Call Stack	515
Postępowanie w przypadku pętli nieskończonych	516
Obsługa błędów wykonania	516
Kiedy trzeba napisać blok obsługi błędów?	516
Przechwytywanie błędów	517
Wyłączanie przechwytywania błędów	519
Wznawianie działania procedury po wystąpieniu błędu	519
Odczytywanie opisu błędu	522
Zgłaszanie własnych błędów	523
Wyłączanie ostrzeżeń	523
Obsługa przerywania makr przez użytkownika w programach Word i Excel	524
Wyłączenie wprowadzania danych przez użytkownika podczas działania procedury ...	524
Wyłączenie możliwości wprowadzania danych przez użytkownika podczas działania konkretnego fragmentu makra	525
Dokumentowanie kodu	525
Pytania kontrolne i zadania	527
Rozdział 18. Budowanie kodu przyjaznego użytkownikom	529
Czym jest makro przyjazne użytkownikowi?	529
Przywracanie środowiska użytkownika	531
Pozostawienie użytkownika w najlepszej pozycji do kontynuowania pracy	532

Informowanie użytkownika o tym, co się dzieje, podczas działania procedury	533
Operowanie kursorem	535
Wyświetlanie informacji na początku makra	536
Komunikacja z użytkownikiem za pośrednictwem okien komunikatu lub okien dialogowych na koniec działania makra	536
Tworzenie pliku dziennika	536
Dbanie o to, aby makro było uruchomione w odpowiednich warunkach	540
Sprzątanie po wykonaniu procedury	540
Cofanie zmian wprowadzonych przez makro	540
Usuwanie tymczasowych plików i folderów	541
Pytania kontrolne i zadania	542
Rozdział 19. Przegląd mechanizmów zabezpieczeń języka VBA	543
W jaki sposób VBA implementuje zabezpieczenia?	543
Podpisywanie projektów makr podpisem cyfrowym	547
Co to jest certyfikat cyfrowy?	548
Zdobywanie certyfikatu cyfrowego	548
Wybór odpowiedniego poziomu zabezpieczeń	557
Zagrożenia bezpieczeństwa stwarzane przez VBA	557
Ochrona przed wirusami w postaci makr	557
Określanie odpowiedniego poziomu zabezpieczeń	558
Dodatkowe mechanizmy zabezpieczeń pakietu Office	559
Ustawienia blokowania plików	560
Blokowanie kodu	561
Pytania kontrolne i zadania	563

Część VI • Programowanie aplikacji pakietu Office 565

Rozdział 20. Model obiektowy Worda i jego najważniejsze obiekty	567
Model obiektowy Worda	567
Korzystanie z kolekcji Documents oraz obiektu Document	570
Tworzenie dokumentu	570
Tworzenie szablonu	571
Zapisywanie dokumentu	572
Zapisywanie dokumentu po raz pierwszy lub zapisywanie go w innym pliku	572
Zapisywanie dokumentu, który był już wcześniej zapisany	576
Zapisywanie wszystkich otwartych dokumentów	576
Sprawdzanie, czy dokument zawiera niezapisane zmiany	577
Otwieranie dokumentu	577
Zamykanie dokumentu	580
Modyfikowanie szablonu dokumentu	581
Drukowanie dokumentu	581
Korzystanie z obiektu ActiveDocument	583
Korzystanie z obiektu Selection	584
Sprawdzanie rodzaju zaznaczenia	584
Sprawdzanie typu „opowieści” zaznaczenia	585
Uzyskiwanie innych informacji na temat bieżącego zaznaczenia	586
Wstawianie tekstu w miejscu zaznaczenia	589

Wstawianie akapitu w zaznaczeniu	590
Stosowanie stylu	590
Rozszerzanie istniejącego zaznaczenia	590
Redukcja istniejącego zaznaczenia	591
Tworzenie zakresów i korzystanie z nich	592
Definiowanie zakresu nazwanego	592
Zmiana definicji zakresu	593
Korzystanie z właściwości Duplicate w celu przechowywania lub kopiowania formatowania	593
Manipulowanie opcjami	594
Sprawdzanie, czy hiperłącza wymagają kliknięć z klawiszem Ctrl	594
Wyłączanie trybu zastępowania	594
Ustawianie domyślnej ścieżki pliku	594
Wyłączanie śledzenia zmian	595
Pytania kontrolne i zadania	596
Rozdział 21. Korzystanie z popularnych obiektów Worda	597
Korzystanie z mechanizmu wyszukiwania z zastępowaniem z poziomu VBA	597
Składnia metody Execute	598
Przykład użycia operacji znajdowania i zamieniania	602
Korzystanie z nagłówków, stopek i numerów stron	603
Jak VBA implementuje nagłówki i stopki?	603
Dostęp do nagłówka lub stopki	603
Sprawdzanie, czy nagłówek lub stopka istnieją	604
Odwoływanie się do nagłówka lub stopki w poprzedniej sekcji	604
Tworzenie innego nagłówka pierwszej strony	604
Tworzenie różnych nagłówków dla stron nieparzystych i parzystych	605
Dodawanie numerów stron do nagłówków i stopek	605
Korzystanie z sekcji, ustawień strony, okien i widoków	608
Dodawanie sekcji do dokumentu	608
Zmiana ustawień strony	609
Otwieranie nowego okna zawierającego otwarty dokument	609
Zamykanie wszystkich okien dokumentu z wyjątkiem pierwszego	610
Podział okna	610
Wyświetlanie planu dokumentu dla okna	611
Przewijanie okna	611
Układanie okien	611
Pozycjonowanie i określanie rozmiaru okna	612
Wyświetlanie elementu w oknie po jego otwarciu	612
Zmiana widoku dokumentu	613
Przełączanie do trybu czytania	613
Powiększanie i zmniejszanie widoku w celu wyświetlania wielu stron	613
Przetwarzanie tabel	613
Tworzenie tabeli	614
Zaznaczanie tabeli	615
Konwersja tekstu na tabelę	615
Zapewnienie umiejscowienia zaznaczenia wewnątrz tabeli	617
Sprawdzanie, czy zaznaczenie znajduje się wewnątrz tabeli	617

Sortowanie tabeli	618
Dodawanie kolumny do tabeli	619
Usuwanie kolumny z tabeli	619
Ustawianie szerokości kolumny	620
Zaznaczanie kolumny	621
Dodawanie wiersza do tabeli	621
Usuwanie wiersza z tabeli	621
Ustawianie wysokości jednego lub większej liczby wierszy	621
Zaznaczanie wiersza	622
Wstawianie komórki	622
Zwracanie tekstu w komórce	622
Wprowadzanie tekstu w komórce	623
Usuwanie komórek	623
Zaznaczanie zakresu komórek	624
Konwersja tabeli lub wierszy na tekst	624
Pytania kontrolne i zadania	626
Rozdział 22. Model obiektowy Excela i jego najważniejsze obiekty	627
Przegląd modelu obiektowego Excela	627
Obiekty kreowane Excela	629
Zarządzanie skoroszytami	629
Tworzenie skoroszytu	629
Zapisywanie skoroszytu	631
Dostęp do chmury	633
Otwieranie skoroszytu	634
Zamykanie skoroszytu	636
Współdzielenie skoroszytu	637
Ochrona skoroszytu	637
Korzystanie z obiektu ActiveWorkbook	638
Praca z arkuszami	638
Wstawianie arkusza	638
Usuwanie arkusza	639
Kopiowanie lub przenoszenie arkusza	640
Drukowanie arkusza	640
Zabezpieczanie arkusza	642
Korzystanie z obiektu ActiveSheet	643
Korzystanie z aktywnej komórki lub zaznaczenia	643
Praca z aktywną komórką	643
Praca z zaznaczeniem	645
Praca z zakresami	645
Praca z zakresem komórek	646
Definiowanie zakresów nazwanych	646
Usuwanie nazwanego zakresu	647
Praca z nazwanym zakresem	647
Praca z zakresem wykorzystanym	647
Korzystanie z komórek specjalnych	647
Wprowadzanie formuły w komórce	648

Ustawianie opcji	649
Ustawianie opcji w obiekcie Application	649
Ustawianie opcji w obiekcie Workbook	650
Pytania kontrolne i zadania	651
Rozdział 23. Korzystanie z popularnych obiektów Excela	653
Praca z wykresami	653
Tworzenie wykresu	653
Określanie danych źródłowych dla wykresu	655
Określanie typu wykresu	655
Korzystanie z serii na wykresie	655
Dodawanie legendy do wykresu	658
Dodawanie tytułu wykresu	658
Praca z osiami wykresu	658
Formatowanie nagłówków i stopek	659
Praca z obiektami Windows	659
Otwieranie nowego okna w skoroszybie	660
Zamykanie okna	660
Aktywacja okna	660
Rozmieszczanie okien i zmiana ich rozmiaru	660
Powiększanie okna i ustawianie opcji wyświetlania	662
Korzystanie z mechanizmu znajdowania i zamieniania	662
Wyszukiwanie za pomocą metody Find	662
Kontynuowanie wyszukiwania z wykorzystaniem metod FindNext i FindPrevious	664
Zamienianie za pomocą metody Replace	664
Wyszukiwanie i zamiana formatowania	664
Dodawanie kształtów	665
Pytania kontrolne i zadania	665
Rozdział 24. Model obiektowy PowerPointa i jego najważniejsze obiekty	667
Przegląd modelu obiektowego PowerPointa	667
Obiekty kreowane programu PowerPoint	669
Praca z prezentacjami	669
Tworzenie nowej prezentacji na podstawie domyślnego szablonu	669
Otwieranie istniejącej prezentacji	670
Otwieranie prezentacji z chmury	671
Zapisywanie prezentacji	671
Zamykanie prezentacji	673
Eksportowanie prezentacji lub wybranych slajdów do formatu graficznego	673
Drukowanie prezentacji	674
Praca z aktywną prezentacją	675
Praca z oknami i widokami	675
Praca z aktywnym oknem	675
Otwieranie nowego okna dla prezentacji	676
Zamykanie okna	676
Aktywacja okna	677
Rozmieszczanie okien na ekranie i zmiana ich rozmiaru	677
Zmiana widoku	677
Praca z panelami okien	677

Praca ze slajdami	678
Dodawanie slajdów do prezentacji	678
Wstawianie slajdów z istniejącej prezentacji	679
Znajdowanie slajdów według numerów ID	680
Zmiana układu istniejącego slajdu	680
Usuwanie istniejącego slajdu	680
Kopiowanie i wklejanie slajdów	681
Powielanie slajdów	681
Przenoszenie slajdów	681
Dostęp do slajdów według nazwy	681
Praca z zakresem slajdów	681
Formatowanie slajdu	682
Ustawianie przejść dla slajdu, grupy slajdów lub wzorca	683
Praca z wzorcami slajdów	684
Praca z wzorcem slajdów	684
Praca z wzorcem tytułu	685
Praca z wzorcem materiałów informacyjnych	685
Praca z wzorcem notatek	685
Usuwanie wzorca	685
Pytania kontrolne i zadania	686
Rozdział 25. Wykorzystywanie kształtów i uruchamianie pokazów slajdów	687
Korzystanie z kształtów	687
Dodawanie kształtów do slajdów	687
Usuwanie kształtu	692
Zaznaczanie wszystkich kształtów	692
Zmiana położenia i rozmiaru kształtu	693
Kopiowanie formatowania z jednego kształtu do innego	693
Posługiwanie się tekstem wewnątrz kształtu	694
Animowanie kształtu lub grupy kształtów	697
Operacje na nagłówkach i stopkach	699
Uzyskanie dostępu do potrzebnego obiektu Header lub Footer	700
Wyświetlanie lub ukrywanie obiektu Header lub Footer	700
Ustawianie tekstu w nagłówku lub stopce	700
Ustawianie formatu daty i godziny nagłówków i stopek	700
Konfigurowanie i uruchamianie pokazu slajdów	701
Zarządzanie typem pokazu	702
Tworzenie pokazu niestandardowego	702
Usuwanie niestandardowego pokazu	703
Uruchamianie pokazu slajdów	703
Zmiana rozmiaru i położenia pokazu slajdów	704
Poruszanie się pomiędzy slajdami	704
Wstrzymywanie pokazu slajdów i korzystanie z białych i czarnych ekranów	704
Uruchamianie i zatrzymywanie pokazów niestandardowych	705
Kończenie pokazu slajdów	705
Pytania kontrolne i zadania	705

Rozdział 26. Model obiektowy Outlooka i jego najważniejsze obiekty	707
Przegląd modelu obiektowego Outlooka	707
Model obiektowy Outlooka	708
Gdzie Outlook przechowuje makra VBA?	709
Najpopularniejsze obiekty kreowalne Outlooka	710
Korzystanie z obiektu Application	710
Wprowadzenie do obiektu NameSpace	711
Praca z przestrzeniami nazw	711
Inspektory i eksploratory	712
Tworzenie elementów programu Outlook	714
Kończenie pracy Outlooka	715
Uniwersalne metody do pracy z obiektami Outlooka	715
Metoda Display	715
Korzystanie z metody Close	716
Korzystanie z metody PrintOut	716
Korzystanie z metody Save	716
Korzystanie z metody SaveAs	717
Operacje na wiadomościach	718
Tworzenie nowej wiadomości	718
Operacje na treści wiadomości	718
Dodawanie załącznika do wiadomości	719
Wysyłanie wiadomości	720
Operacje na elementach kalendarza	721
Tworzenie nowego elementu kalendarza	721
Operacje na treści elementu kalendarza	721
Postępowanie się zadaniami i zleceniami zadań	722
Tworzenie zadania	722
Operacje na zawartości elementu zadania	723
Przypisywanie zadania do współpracownika	723
Wyszukiwanie informacji	724
Pytania kontrolne i zadania	726
Rozdział 27. Zdarzenia w Outlooku	727
Czym różni się procedury obsługi zdarzeń od zwykłych makr?	727
Korzystanie ze zdarzeń poziomu aplikacji	728
Korzystanie ze zdarzenia Startup	730
Korzystanie ze zdarzenia Quit	730
Korzystanie ze zdarzenia ItemSend	731
Korzystanie ze zdarzenia NewMail	732
Korzystanie ze zdarzeń AdvancedSearchComplete i AdvancedSearchStopped	732
Korzystanie ze zdarzenia MAPILogonComplete	733
Zdarzenie Reminder	733
Zdarzenie OptionsPagesAdd	733
Zdarzenia na poziomie elementów	734
Deklarowanie zmiennej obiektowej i inicjowanie zdarzenia	735
Zdarzenia, które mają zastosowanie do wszystkich elementów reprezentujących wiadomości	735
Zdarzenia dotyczące eksploratorów, inspektorów i widoków	737

Zdarzenia dotyczące folderów	741
Zdarzenia dotyczące obiektów Items i Results	741
Zdarzenia dotyczące przypomnień	741
Zdarzenia dotyczące synchronizacji	741
Szybkie kroki	742
Pytania kontrolne i zadania	744
Rozdział 28. Model obiektowy Accessa i jego najważniejsze obiekty	745
Pierwsze kroki z VBA w programie Access	746
Tworzenie modułu w VBE	748
Tworzenie funkcji	748
Budowniczy makr	748
Tworzenie makra w stylu Accessa w celu wywołania funkcji	748
Tłumaczenie makr w stylu Accessa na makra VBA	750
Korzystanie z makra Autoexec w celu zainicjowania sesji Accessa	751
Uruchamianie procedur Sub	752
Instrukcja Option Compare Database	752
Przegląd modelu obiektowego Accessa	753
Obiekty kreowane Accessa	753
Otwieranie i zamykanie baz danych	755
Korzystanie z metody CurrentDb w celu uzyskania dostępu do bieżącej bazy danych	755
Zamykanie bieżącej bazy danych i otwieranie innej bazy danych	755
Komunikacja między aplikacjami pakietu Office	758
Otwieranie wielu baz danych naraz	759
Zamykanie bazy danych	761
Tworzenie i usuwanie obszarów roboczych	761
Korzystanie z obiektu Screen	762
Korzystanie z obiektu DoCmd do uruchamiania poleceń Accessa	763
Wykorzystanie metody OpenForm do otwierania formularzy	766
Wykorzystanie metody PrintOut do drukowania obiektu	766
Korzystanie z metody RunMacro do uruchamiania makr w stylu Accessa	767
Pytania kontrolne i zadania	768
Rozdział 29. Dostęp do aplikacji z innej aplikacji	769
Narzędzia do komunikacji pomiędzy aplikacjami	769
Zastosowanie Automatykacji do przesyłania informacji	770
Wiązanie wczesne i późne	771
Tworzenie obiektu za pomocą funkcji CreateObject	772
Zwracanie obiektu za pomocą funkcji GetObject	772
Przykłady wykorzystania Automatykacji dla aplikacji pakietu Office	773
Korzystanie z funkcji Shell do uruchamiania aplikacji	782
Korzystanie z obiektów danych do przechowywania i pobierania informacji	784
Tworzenie obiektu danych	785
Zapisywanie informacji w obiekcie danych	785
Pobieranie informacji z obiektu danych	786
Umieszczanie informacji w Schowku	786
Sprawdzanie, czy obiekt danych zawiera określony format	787

Komunikacja za pośrednictwem DDE	787
Użycie metody DDEInitiate do nawiązania połączenia DDE	788
Korzystanie z metody DDERequest w celu odczytania tekstu z innej aplikacji	788
Korzystanie z metody DDEPoke w celu wysłania tekstu do innej aplikacji	789
Wykorzystanie metody DDEExecute do uruchamiania polecenia jednej aplikacji z innej aplikacji	790
Zamykanie kanału DDE za pomocą metody DDETerminate	790
Zamykanie wszystkich otwartych kanałów DDE za pomocą metody DDETerminateAll	790
Komunikacja za pomocą mechanizmu SendKeys	791
Nie tylko VBA	795
Pytania kontrolne i zadania	796
Dodatek. Odpowiedzi i rozwiązania zadań	797
Rozdział 1. Rejestrowanie i uruchamianie makr w aplikacjach pakietu Office	797
Rozdział 2. Wprowadzenie do programu Visual Basic Editor	799
Rozdział 3. Edycja zarejestrowanych makr	800
Rozdział 4. Tworzenie kodu w edytorze VBE od podstaw	800
Rozdział 5. Podstawy składni VBA	803
Rozdział 6. Zmienne, stałe i wyliczenia	804
Rozdział 7. Zmienne tablicowe	806
Rozdział 8. Wyszukiwanie obiektów, metod i właściwości, których potrzebujesz	807
Rozdział 9. Korzystanie z funkcji wbudowanych	808
Rozdział 10. Tworzenie własnych funkcji	809
Rozdział 11. Podejmowanie decyzji w kodzie	811
Rozdział 12. Korzystanie z pętli do powtarzania działań	812
Rozdział 13. Pobieranie danych od użytkownika za pomocą okien MsgBox i InputBox	813
Rozdział 14. Tworzenie prostych, niestandardowych okien dialogowych	814
Rozdział 15. Tworzenie złożonych formularzy	817
Rozdział 16. Budowanie modułowego kodu i korzystanie z klas	819
Rozdział 17. Debugowanie kodu i obsługa błędów	820
Rozdział 18. Budowanie kodu przyjaznego użytkownikom	822
Rozdział 19. Przegląd mechanizmów zabezpieczeń języka VBA	823
Rozdział 20. Model obiektowy Worda i jego najważniejsze obiekty	825
Rozdział 21. Korzystanie z popularnych obiektów Worda	826
Rozdział 22. Model obiektowy Excela i jego najważniejsze obiekty	827
Rozdział 23. Korzystanie z popularnych obiektów Excela	828
Rozdział 24. Model obiektowy PowerPointa i jego najważniejsze obiekty	829
Rozdział 25. Wykorzystywanie kształtów i uruchamianie pokazów slajdów	830
Rozdział 26. Model obiektowy Outlooka i jego najważniejsze obiekty	831
Rozdział 27. Zdarzenia w Outlooku	831
Rozdział 28. Model obiektowy Accessa i jego najważniejsze obiekty	832
Rozdział 29. Dostęp do jednej aplikacji z innej aplikacji	833

Rozdział 1.

Rejestrowanie i uruchamianie makr w aplikacjach pakietu Office

W pierwszym rozdziale odkryjesz najprostszy sposób rozpoczęcia pracy z językiem Visual Basic for Applications (VBA): nagrywanie prostych *makr* za pomocą Rejestratora makr, wbudowanego narzędzia w aplikacji Office. Następnie zobaczysz, jak uruchamiać makra w celu wykonywania przydatnych zadań.

Termin makro zdefiniuję za chwilę. Teraz po prostu zapamiętaj, że dzięki rejestrowaniu *makr* można zautomatyzować proste, ale żmudne powtarzalne zadania i przyspieszyć swoją zasadniczą pracę. Możesz również użyć Rejestratora makr do tworzenia kodu VBA, który wykonuje potrzebne działania, a następnie możesz zmodyfikować kod — zwiększyć jego elastyczność i możliwości. Język VBA jest prawdziwą potęgą, jeśli wiemy, jak go używać. Ta książka pokazuje, jak skorzystać z tej mocy.

W TYM ROZDZIALE NAUCZYSZ SIĘ, JAK:

- Zarejestrować makro.
- Przypisać makro do przycisku lub skrótu klawiaturowego.
- Uruchomić makro.
- Usunąć makro.

Czym jest VBA i co można zrobić za jego pomocą?

Visual Basic for Applications jest stworzonym przez Microsoft językiem programowania, który jest wbudowany w aplikacjach Office. Język VBA służy do automatyzacji operacji we wszystkich głównych aplikacjach pakietu Office — Wordzie, Excelu, Outlooku, Accessie i PowerPoincie.

Proszę jednak nie zniechęcać się z powodu twierdzenia, że będziesz *programować*. Wkrótce przekonasz się, że praca z językiem VBA jest prawie zawsze bardzo prosta. Często nie musisz samodzielnie pisać żadnego kodu w VBA. Wystarczy zarejestrować makro, a aplikacja pakietu Office „napisze” cały kod VBA za Ciebie.

Pojęcie automatyzowania operacji w aplikacjach może być dla Czytelnika dość abstrakcyjne. Oto kilka przykładów, jak używać języka VBA w celu usprawnienia zadań, uniknięcia uciążliwego powtarzania operacji, personalizowania interfejsu aplikacji i ogólnej poprawy wydajności:

- Możesz nagrać makro automatycznie wykonujące szereg czynności, które często wykonujesz. Powiedzmy, że często edytujesz dokumenty Worda napisane przez współpracownika, ale on ustawia stopień powiększenia na 100%. Wolisz używać powiększenia 150%. Wszystkim, czego potrzebujesz, aby automatycznie zmienić poziom powiększenia, jest następujący kod VBA:

```
ActiveWindow.ActivePane.View.Zoom.Percentage = 150
```

Możesz nawet umieścić ten kod w specjalnym miejscu w Wordzie, dzięki czemu zmodyfikowane powiększenie zostanie zastosowane dla każdego otwieranego dokumentu:

```
Sub AutoOpen()  
    ActiveWindow.ActivePane.View.Zoom.Percentage = 150  
End Sub
```

Nie martw się. Nie musisz nawet znać takich terminów programowania, jak `ActiveWindow` czy `View.Zoom`. Wystarczy, że włączysz rejestrator makr. Następnie klikniesz *Widok*, potem *Powiększenie* i ustawisz 150%. Rejestrator będzie „obserwować” kroki, które wykonujesz, a następnie wygeneruje niezbędny kod VBA, który potrafi odtworzyć te kroki. W ogóle nie musisz pisać kodu!

- Możesz napisać kod, który wykonuje czynności pewną liczbę razy oraz podejmuje decyzje w zależności od sytuacji, w której jest uruchomiony. Na przykład może wykonać szereg operacji na każdej prezentacji otwieranej w programie PowerPoint.
- Możesz użyć VBA do zmodyfikowania wyglądu lub zachowania interfejsu użytkownika. VBA może, na przykład, komunikować się z użytkownikiem poprzez wyświetlanie formularzy lub niestandardowych okien dialogowych, pozwalających mu dokonać wyboru i określić ustawienia. Możesz wyświetlić zestaw opcji formatowania za pomocą kontroltek, takich jak pola wyboru i przyciski opcji, które użytkownik może zaznaczyć. Następnie, kiedy zamknie on okno dialogowe, makro podejmuje odpowiednie działania na podstawie wprowadzonych przez niego danych.
- Za pośrednictwem VBA możesz zrealizować działania, których nie można łatwo wykonać w przypadku ręcznego posługiwania się interfejsem użytkownika lub które wcale nie są dostępne ręcznie. Na przykład, w trybie interaktywnym, w większości aplikacji jesteś ograniczony do pracy z aktywnym plikiem — aktywnym dokumentem w programie Word, aktywnym skoroszytem w programie Excel i tak dalej. Dzięki użyciu VBA, możesz uzyskać dostęp i zarządzać plikami, które *nie są* aktywne.
- Możesz zarządzać jedną aplikacją z poziomu innej aplikacji. Na przykład, możesz z poziomu Worda umieścić tabelę z dokumentu Worda w arkuszu Excela.

Takie zadania — i wiele innych — zostaną omówione w tej książce.

Różnica między językami Visual Basic a Visual Basic for Applications

VBA bazuje na Visual Basic — języku programowania wywodzącym się od języka BASIC. Język BASIC powstał w 1963 roku, a jego nazwa pochodzi od *Beginner's All-Purpose Symbolic Instruction Code* — dosł. kod instrukcji ogólnego przeznaczenia dla początkujących. BASIC został zaprojektowany w sposób przyjazny dla użytkownika, ponieważ korzysta z rozpoznawalnych słów angielskich (lub ich odmian), zamiast zawiłych i niezrozumiałych terminów programistycznych, występujących w takich językach jak C. Oprócz zastosowania angielskiej „dykcji” projektanci BASICA starali się zapewnić jak najprostszą interpunkcję i składnię języka.

Visual Basic jest wizualny, ponieważ oferuje wydajne skróty, takie jak programowanie za pomocą techniki przeciągnij i upuść, z wykorzystaniem wielu elementów graficznych.

Pomimo swej prostoty, czyniącej go językiem przyjaznym programistom, VB jest także mocny i wydajny, jak dowolny inny język programowania!

Visual Basic for Applications — odmiana języka VB, której jest poświęcona ta książka — to wersja Visual Basica dostosowana do zarządzania aplikacjami pakietu Microsoft Office.

Każda aplikacja Office ma swoją własną kolekcję *obiektów* (cech i zachowań). Ich zbiory dostępne w każdej z aplikacji są nieco inne, bo nie ma dwóch aplikacji, które współdzielałoby dokładnie te same własności i polecenia.

Na przykład niektóre obiekty VBA dostępne w Wordzie nie są dostępne w Excelu (i vice versa), ponieważ kilka narzędzi Worda, choćby generator spisu treści, nie jest prawidłowych w Excelu.

Istnieje jednak obszerny zbiór podstawowych poleceń, struktur i technik programowania w VBA, które są identyczne w Wordzie i Excelu. Jak się przekonasz, często dość łatwo można przełożyć wiedzę na temat programowania w VBA w Wordzie na programowanie w VBA w Excelu (lub VBA jakiegokolwiek aplikacji, która obsługuje ten język).

Na przykład metody *Save* (*metoda* jest w istocie działaniem, które można wykonać) można użyć, aby zapisać plik w VBA Excela, Worda lub PowerPointa. Różne są zapisywane *obiekty*. W VBA Excela odpowiednią komendą będzie `ActiveWorkbook.Save`, w VBA Worda będzie to komenda `ActiveDocument.Save`, natomiast w VBA PowerPointa `ActivePresentation.Save`.

VBA zawsze działa w aplikacji hosta (takiej jak Access lub Word). Z wyjątkiem nielicznych autonomicznych programów, które zazwyczaj najlepiej stworzyć za pomocą Visual Basic .NET, w celu uruchomienia kodu VBA zawsze musi być otwarta aplikacja hosta. Oznacza to, że nie można zbudować w VBA samodzielnych aplikacji w sposób, w jaki można to zrobić w Visual Basicu. Jeśli chcesz, możesz ukryć aplikację hosta przed użytkownikami w taki sposób, aby widzieli wyłącznie interfejs (zazwyczaj formularze użytkownika) utworzony dla Twoich procedur w VBA. W ten sposób można stworzyć iluzję samodzielnych aplikacji. Język VBA jest jednak rzadko wykorzystywany w tym celu. Jeśli chcesz pisać samodzielne programy, zapoznaj się raczej z Visual Basic Express.

Co to jest Visual Basic .NET i Visual Basic Express?

Visual Basic .NET (VB .NET) to tylko jedna z wersji w długiej historii implementacji języka BASIC firmy Microsoft. BASIC obejmuje obszerny zestaw gotowych bibliotek, które pozwalają zrobić prawie wszystko, co jest możliwe w systemie Windows. Chociaż VB .NET jest powszechnie stosowany do pisania autonomicznych aplikacji, można wykorzystać jego biblioteki, jeśli zachodzi taka potrzeba, wewnątrz makr VBA.

Należy zapamiętać, że każda aplikacja pakietu Office posiada własną bibliotekę obiektów, lecz biblioteki .NET zawierają wiele dodatkowych własności (często wykonujących operacje systemu operacyjnego Windows). A zatem, jeśli potrzebujesz własności, której nie możesz znaleźć wewnątrz VBA lub w bibliotece obiektów aplikacji Office, możesz skorzystać z zasobów całej biblioteki .NET.

Visual Basic Express to darmowa wersja języka VB .NET. Po zapoznaniu się z VBA w tej książce, można pobrać i zapoznać się z Visual Studio Express for Desktop. Archiwum jest dostępne pod adresem: <https://visualstudio.microsoft.com/pl/vs/express>.

Można skorzystać z wersji *Community* lub przewinąć stronę w dół, aby znaleźć wersję Express. Jeśli jesteś zainteresowany operacjami w samym systemie Windows, możesz zapoznać się z produktem AutoHotkey. To potężny zasób dla tych, którzy chcą mieć większą kontrolę nad swoim komputerem: <https://www.autohotkey.com/>.

Podstawowe wiadomości o makrach

Makro to ciąg poleceń, które mogą być wykonane w dowolnym momencie. Taka definicja pasuje również do *programu komputerowego*. Makra są jednak na ogół krótkimi programami dedykowanymi do pojedynczego zadania. Pomyśl o tym w następujący sposób: zwykły program komputerowy, taki jak Photoshop czy Chrome, ma wiele możliwości. Chrome może zapisywać linki do ulubionych stron, pokazać źródło dowolnej strony internetowej (*Ctrl+Shift+I*), blokować strony internetowe, wyświetlać je na pełnym ekranie po naciśnięciu klawisza *F11* i tak dalej.

Makro jest mniejsze, dedykowane do realizacji tylko jednego z tych zadań, na przykład wyświetlania strony na pełnym ekranie. Zatem makro prawdopodobnie doda jedną nową funkcjonalność do ogromnego zbioru funkcjonalności już wbudowanych w aplikację pakietu Office.

Niektóre aplikacje pozwalają na takie skonfigurowanie makr, aby uruchamiały się automatycznie. Na przykład, można utworzyć makro w programie Word, które automatycznie stosuje podstawowe zadania formatowania dla dokumentów, które regularnie otrzymujesz nieprawidłowo sformatowane. Jak zobaczysz w rozdziale 6., „Wykorzystanie zmiennych, stałych i typów wyliczeniowych”, przy okazji opisu funkcji *Autoexec*, można sprawić, aby makro uruchomiło się automatycznie po otwarciu dokumentu tego typu.

Makro jest rodzajem *podprogramu* (czasami nazywanego również *podprocedurą* lub *funkcją*).

Jak wkrótce się dowiesz, Visual Basic Editor rozpoczyna każde z makr słowem kluczowym *Sub* lub *Func* i *on*. Zapamiętaj więc, że makro jest pojedynczą procedurą, podczas gdy program komputerowy, taki jak Photoshop lub Word, zawiera zbiór wielu procedur.

W aplikacji pakietu Office, która obsługuje Rejestrator makr VBA (Word lub Excel), można tworzyć makra na dwa sposoby:

- Włącz Rejestrator makr i po prostu wykonaj ręcznie sekwencję działań, które ma wykonywać makro. Możesz klikać, pisać, przeciągać i upuszczać — wszystko, co zrobisz, będzie zarejestrowane.
- Otwórz Visual Basic Editor i wpisz w jego oknie polecenia VBA, aby napisać makro bez wcześniejszego rejestrowania.

Istnieje również przydatne podejście hybrydowe, które łączy rejestrowanie z edycją. Najpierw zarejestruj sekwencję działań, a następnie przejrzyj zarejestrowane makro w programie Visual Basic Editor i ewentualnie je zmodyfikuj. Możesz w ten sposób usunąć wszystkie niepotrzebne polecenia lub dodać nowe. Możesz też skorzystać z przybornika dostępnego w edytorze, aby przeciągać i upuszczać do makra elementy interfejsu użytkownika (takie jak okna komunikatów i okna dialogowe). Dzięki nim użytkownicy mogą podejmować decyzje i wybierać opcje określające sposób uruchomienia makr. Makra są niezwykle elastyczne, a edytor VBA jest potężnym, ale prostym w obsłudze narzędziem. Edytor jest dla programowania tym, czym Word jest dla pisania — bardzo dojrzałym, wydajnym i dobrze zaprojektowanym zbiorem narzędzi.

Gdy już utworzysz makro, musisz określić, w jaki sposób chcesz pozwolić użytkownikowi je uruchomić. W większości przypadków możesz przypisać makro do Wstążki, paska narzędzi *Szybki dostęp* lub kombinacji klawiszy skrótu. Dzięki temu uruchomienie makra jest bardzo łatwe — sprowadza się do kliknięcia ikony lub naciśnięcia klawisza skrótu (na przykład *Alt+R*). Można też opcjonalnie przypisać makro do przycisku na pasku *Szybki dostęp* lub skrótu klawiatury przy pierwszym rejestrowaniu makra; zrobisz to za pomocą okna dialogowego, które pojawia się automatycznie, kiedy rozpoczniesz rejestrowanie.

Wkrótce dowiesz się, jak to wszystko działa. To proste. Aby przypisać makro do Wstążki, najpierw je zarejestruj, a następnie kliknij prawym przyciskiem myszy Wstążkę i wybierz *Dostosuj Wstążkę*. Na rozwijanej liście kliknij przycisk *Wybierz polecenia*, a potem pozycję *Makra*, aby wyświetlić wszystkie makra.

Rejestrowanie makr

Najprostszym sposobem utworzenia kodu VBA jest zarejestrowanie makra przy użyciu Rejestratora makr. Narzędzie to jest dostępne tylko w programach Word i Excel.

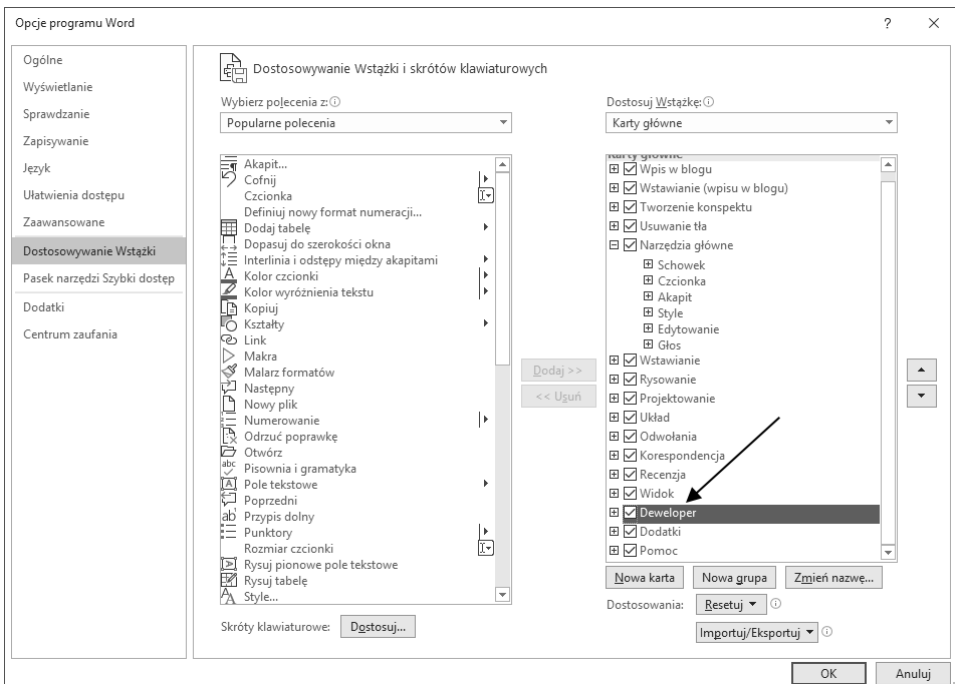
Należy włączyć Rejestrator makr i ewentualnie przypisać wyzwalacz (przycisk paska narzędzi lub kombinację klawiszy skrótów), który pozwoli później uruchomić makro. Następnie trzeba wykonać czynności, które ma wykonywać makro, a na koniec wyłączyć rejestrator. Podczas wykonywania czynności Rejestrator makr tłumaczy je na komendy — *kod* — w języku programowania VBA.

Po zakończeniu rejestrowania makra można wyświetlić kod w programie Visual Basic Editor i zmodyfikować go, jeśli jest taka potrzeba. Jeśli po zarejestrowaniu kod działa dobrze, nie musisz do niego zaglądać. Możesz po prostu w dowolnym momencie uruchomić makro, klikając odpowiedni przycisk paska narzędzi lub wciskając kombinację klawiszy przypisaną do makra.

Wyświetlanie zakładki Deweloper na Wstążce

Zanim przejdziemy dalej, zadaj o to, aby na Wstążce była widoczna zakładka *Deweloper* (programista). Ta zakładka jest bramą do makr VBA i edytora VBA. Domyślnie Microsoft nie wyświetla tej opcji, aby niepotrzebnie nie mylić użytkowników niebędących programistami (w programach Access i OneNote nawet nie ma tej zakładki, jest natomiast w programach Word, Excel, PowerPoint i Outlook). Ale ponieważ Ty *jestes* programistą, to powinieneś dodać zakładkę *Deweloper* na swojej Wstążce (patrz rysunek 1.1):

1. Kliknij najpierw zakładkę *Plik*, a następnie polecenie *Opcje*.
Polecenie *Opcje* zostało przeniesione do dolnej lewej części ekranu.
2. Kliknij przycisk *Dostosowywanie Wstążki*.
3. Przewiń w dół pole listy po prawej stronie i wybierz opcję *Deweloper*, aby ją zaznaczyć.
4. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Opcje*.



RYSUNEK 1.1. Kliknij tutaj, aby dodać swoją zakładkę Deweloper

Teraz zobaczysz na swojej Wstążce nową zakładkę Deweloper, znajdującą się po prawej stronie zakładek domyślnych.

W kolejnych punktach przyjrzymy się czynnościom wykonywanym podczas rejestrowania makra. Proces jest łatwy, ale użytkownicy, którzy nie robili tego wcześniej, potrzebują odrobinę wprowadzenia. Po zamieszczeniu ogólnego opisu zarejestrujemy przykładowe makra w Wordzie i Excelu (w dalszej części książki, kiedy nauczysz się korzystać z programu Visual Basic Editor, przeanalizujemy i zmodyfikujemy te makra, więc nie usuwaj ich).

Planowanie makra

Jeszcze przed uruchomieniem Rejestratora makr warto trochę poplanować. Pomyśl o tym, co będzie robić makro. W większości przypadków można po prostu zarejestrować je i nie martwić się *kontekstem*. Można po prostu je nagrać z otwartego dokumentu.

Jednak w niektórych sytuacjach przed rozpoczęciem nagrywania należy zadbać o skonfigurowanie specjalnego kontekstu. Na przykład możesz utworzyć w programie Word makro, które wykonuje pewne zadania związane z edycją tekstu, takie jak wyróżnienie słowa kursywą i podkreślenie go. Aby to zrobić, potrzebny jest najpierw ustawiony przed słowem migający kursor „wstawiania”, który nie jest pochylony lub podkreślony. Nie chcesz zarejestrować działania przesuwania kursora wstawiania do danego słowa. To spowodowałoby, że makro stałoby się specyficzne dla tego dokumentu i tego słowa w tym dokumencie. Zazwyczaj chcemy, aby makro działało z więcej niż jednym konkretnym dokumentem. Twoje makro ma wyróżniać kursywą i podkreśleniem *dowolne* słowo wskazywane przez migający kursor w dowolnym dokumencie.

Niemniej jednak, większość prostych makr można rejestrować bez specjalnego planowania. Wystarczy nagrać cokolwiek chcesz, aby makro robiło.

ZATRZYMYWANIE MAKRA

Word (ale nie Excel) pozwala zatrzymać działanie Rejestratora makr, jeśli musisz się zatrzymać podczas nagrywania, aby zrobić coś, czego nie chcesz zarejestrować. Ta funkcjonalność pozwala poradzić sobie z problemami, których nie przewidziałeś podczas planowania makra – na przykład koniecznością otwarcia dokumentu, który powinien być otwarty przed rozpoczęciem nagrywania makra.

Niektóre zarejestrowane makra generują kod, który sam przeprowadza wszystkie niezbędne ustawienia. Kontekst konfiguracji zostanie zarejestrowany jako część makra. W takich przypadkach, przed rozpoczęciem rejestrowania makra należy zadbać o to, aby aplikacja była w stanie, którego makro oczekuje.

Na przykład, jeżeli makro do wykonania swojego zadania potrzebuje pustego aktywnego skoroszytu w programie Excel, samo powinno utworzyć pusty skoroszyt zamiast używać tego, który akurat jest aktywny w danej chwili. Oszczęda to użytkownikowi wykonywania dodatkowego kroku, kiedy makro działa. A zatem, aby to zapewnić, uruchom rejestrowanie *przed* uruchomieniem pustego aktywnego skoroszytu.

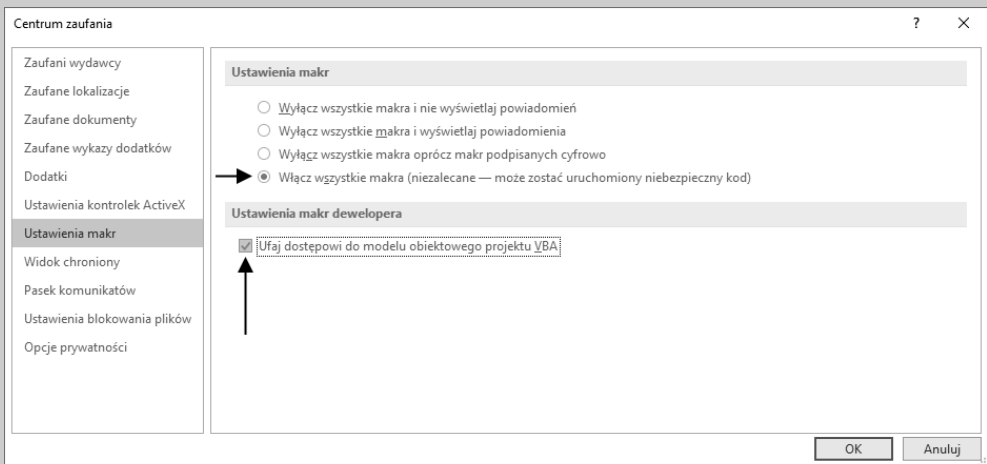
UWAGA NA TEMAT BEZPIECZEŃSTWA

Makra są programami komputerowymi, choć zwykle niewielkimi. Z poziomu makra można nawet skorzystać ze wszystkich funkcji systemu operacyjnego Windows. Z tego powodu w makrze mogą być zawarte wirusy i inny złośliwy kod (taki kod może uruchomić się automatycznie przez samo otwarcie zainfekowanego dokumentu za pomocą funkcji Autoexec, wspomnianej wcześniej w tym rozdziale, a omówionej w rozdziale 6., a także innych technik, takich jak wykorzystanie folderu Autostart aplikacji). Na przykład, jeśli użytkownik otworzy zainfekowany dokument Worda, wirus osadzony w makrze może usuwać pliki na dysku twardym. To jest oczywiście bardzo niebezpieczne.

Aplikacje pakietu Office 2019, a także sam system operacyjny Windows, zawierają wiele warstw zabezpieczeń w celu ochrony przed wirusami i szkodliwym kodem. Specyficzna dla makr jest technologia „zaufania”, która jest wbudowana w aplikacje pakietu Office. Aby wyświetlić lub zmienić ustawienia zaufania, otwórz okno dialogowe *Centrum zaufania*. W tym celu kliknij zakładkę *Deweloper* na Wstążce, a następnie w programach Word, Excel, Outlook lub PowerPoint kliknij ikonę *Bezpieczeństwo makr* (po lewej stronie, w sekcji *Kod* na Wstążce). W Accessie, jak to często bywa, robi się to trochę inaczej niż w innych aplikacjach pakietu Office. Access nie ma zakładki *Deweloper*. Aby zarządzać zabezpieczeniami makr w tym programie, kliknij kolejno zakładkę *Plik*, polecenie *Opcje* po lewej stronie, przycisk *Centrum zaufania*, przycisk *Ustawienia Centrum zaufania*, a na koniec przycisk *Ustawienia makr*.

Chodzi o to, że jeśli nie można uruchamiać makr lub jeśli pojawiają się tajemnicze komunikaty o błędach, takie jak „Nie można utworzyć makra” lub „Odmowa dostępu”, albo na Wstążce opcje makr są szare, nie dają się kliknąć lub są wyłączone, to być może trzeba dokonać pewnych korekt w ustawieniach.

Jeśli takie rzeczy się dzieją, pierwszym krokiem powinno być sprawdzenie *Centrum zaufania* i zaznaczenie opcji *Wyłącz wszystkie makra i wyświetlaj powiadomienia*. To ustawienie powoduje wyświetlanie kierowanych do użytkownika pytań o pozwolenie na uruchomienie makra. Możesz także, na czas pracy z makrami w tej książce, po prostu włączyć w Centrum zaufania opcję *Włącz wszystkie makra*. Następnie usuń zaznaczenie tej opcji przed zamknięciem dokumentu, z którym pracowałeś podczas lektury tej książki. Chodzi o to, że można zaufać własnym makrom, ale nie należy ufać wszystkim makrom ze wszystkich dokumentów, które możesz otrzymać ze źródeł zewnętrznych. Gdy już jesteś w Centrum zaufania, możesz także zaznaczyć opcję *Zaufaj dostępowi do modelu obiektów projektu VBA* – tak jak pokazano na poniższej ilustracji.



Jeśli pracujesz nad dokumentem, który sam utworzyłeś, i który zawiera makra, które sam napisałeś, możesz zaufać temu dokumentowi i zgodzić się na uaktywnienie makr. Jeśli jednak otwierasz dokument pochodzący od kogoś innego, powinieneś zachować ostrożność.

Dodatkowe kwestie bezpieczeństwa można rozwiązać poprzez zarządzanie różnymi warstwami zabezpieczeń, które teraz, z konieczności, są osadzone w systemach operacyjnych i aplikacjach. Jednym ze sposobów radzenia sobie z kwestiami bezpieczeństwa jest analiza zabezpieczeń dostępnych w menu *Pomoc* aplikacji systemów Windows 7, 8 lub 10. Czasami też można uzyskać dobre odpowiedzi, zamieszczając pytania w internetowych grupach użytkowników lub przeszukując eksperckie witryny internetowe, takie jak Wikipedia. Ponadto dobry przegląd zabezpieczeń makr w systemie Office 2019 można znaleźć na stronie: <https://support.office.com/en-us/article/enable-or-disable-macros-in-office-files-12b036fd-d140-4e74-b45e-16fed1a7e5c6>.

Zagadnienia zabezpieczeń pakietu Office 2019 zostały dokładnie opisane w rozdziale 19., „Własności zabezpieczeń VBA”.

Uruchamianie Rejestratora makr

Aby uruchomić Rejestrator makr, kliknij najpierw zakładkę *Deweloper* na Wstążce, a następnie przycisk *Zarejestruj makro* (rysunek 1.2). Możesz także kliknąć przycisk *Zarejestruj makro* na pasku stanu u dołu aplikacji (stosując to podejście, nie musisz otwierać zakładki *Deweloper*; wystarczy, że klikniesz przycisk na pasku stanu). Wygląda on tak:

RYSUNEK 1.2.

Znajdź przycisk *Zarejestruj makro* na pasku stanu



Kiedy uruchomisz Rejestrator makr, otworzy się okno dialogowe *Rejestrowanie makra*. Widać na nim, że nowemu makru nadano domyślną nazwę (*Makro1*, *Makro2* i tak dalej). Możesz zaakceptować tę domyślną nazwę lub ją zmienić. Dostępny jest również opcjonalny opis, który możesz wypełnić, jeśli chcesz.

Aby zatrzymać Rejestrator makr, możesz kliknąć przycisk *Zatrzymaj rejestrowanie* na zakładce *Deweloper*. Alternatywnie możesz zatrzymać rejestrowanie klikając kwadratowy czarny przycisk, który wyświetla się podczas rejestrowania na pasku stanu w lewym dolnym rogu okna aplikacji (jest to czarna ikona rejestrowania).

Po zatrzymaniu rejestratora, kwadratowy przycisk zostanie zastąpiony ikoną, którą możesz kliknąć, aby rozpocząć rejestrowanie nowego makra (w programie Word dla komputerów Mac należy pojedynczo kliknąć wskaźnik REC zamiast klikać go dwukrotnie).

Wygląd okna dialogowego *Rejestrowanie makra* jest nieco różny w Wordzie i Excelu, ponieważ okno dialogowe musi oferować odpowiednie opcje, właściwe dla różnych specyficznych możliwości każdej aplikacji. W każdym przypadku należy nadać makru nazwę i dodać opis. Zazwyczaj można również określić, gdzie zapisać makro, na przykład Word oferuje dwie opcje:

- jeśli makro ma być przeznaczone do użytku globalnego (będzie dostępne we wszystkich dokumentach Worda), należy je zapisać w pliku o nazwie *normal.dotm*;
- jeśli makro będzie używane jedynie w bieżącym aktywnym dokumencie, zapisz je w pliku o nazwie dokumentu z rozszerzeniem *.dotm*.

Zwykły szablon Worda ma rozszerzenie nazwy pliku *.dotx*, ale makra są zapisywane w pliku z rozszerzeniem *.dotm*.

Excel udostępnia trzy opcje: przechowywanie makra w bieżącym skoroszytce lub w nowym skoroszytce, albo przeznaczenie makra do użytku we wszystkich skoroszytach Excela, w skoroszytce makr osobistych. Jest to Excelowy odpowiednik pliku *Normal.dotm* z Worda (skoroszyt makr osobistych Excela zostanie zapisany w pliku o nazwie *Personal.xlsb*). Temu specyficznemu skoroszytowi przyjrzymy się wkrótce.

GDZIE PRZECHOWYWAĆ MAKRA W PROGRAMIE POWERPOINT?

Nie można rejestrować makr w wersji 2019 PowerPointa, ale można je tworzyć poprzez pisanie kodu w programie Visual Basic Editor. Następnie można zapisać makra w bieżącej aktywnej prezentacji lub w dowolnej innej otwartej prezentacji lub szablonie.

PowerPoint zapewnia również globalny kontener makr (podobny do pliku *Normal.dotm* Worda). W programie PowerPoint wybierz opcję *Wszystkie otwarte prezentacje* w polu listy makr. Aby ją znaleźć, kliknij ikonę *Makra* w sekcji *Kod* zakładki *Deweloper* na *Wstążce*.

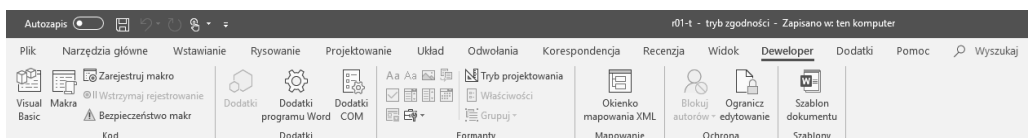
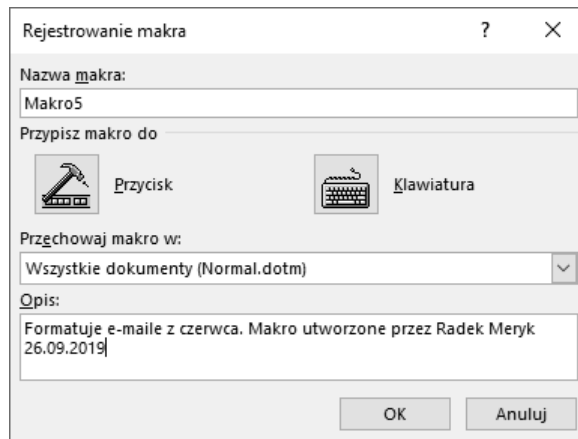
Okno dialogowe *Rejestrowanie makra* pozwala również określić sposób, w jaki chcesz uruchamiać makro. Word wyświetla przyciski, które można kliknąć, aby otworzyć okno dialogowe, wprowadzić kombinację klawiszy skrótów lub otworzyć okno *Opcje programu Word*, gdzie można utworzyć dla makra przycisk, który pojawi się na pasku narzędzi *Szybki dostęp*. Excel ogranicza możliwości uruchamiania makr do zastosowania skrótów klawiszowego *Ctrl+*, więc nie ma przycisku pozwalającego wyświetlić kompletne okno dialogowe skrótów klawiaturowych, takie jak to dostępne w programie Word. W Excelu jest tylko niewielkie pole tekstowe, w którym w celu zdefiniowania skrótów można wprowadzić klawisz tworzący parę z klawiszem *Ctrl*.

Większość tych aplikacji firmy Microsoft, które obsługują VBA, posiada zakładkę *Deweloper*, z której można zarządzać procesem rejestrowania makra, uruchomić edytor Visual Basic oraz zarządzać makrami w inny sposób. Jednak Access grupuje kilka narzędzi związanych z makrami na zakładce *Narzędzia bazy danych* (która jest domyślnie widoczna), a także posiada opcję *Makro* na zakładce *Tworzenie*.

Na rysunku 1.3 pokazano okno dialogowe *Rejestrowanie makr* w programie Word ze standardową nazwą i wprowadzonym opisem. Na rysunku 1.4 przedstawiono wersję zakładki *Deweloper* na *Wstążce* z programu Word.

RYSUNEK 1.3.

W oknie dialogowym *Rejestrowanie makr* wprowadź nazwę makra, które masz zamiar zarejestrować. Możesz też wpisać zwięzły opis w polu *Opis*. Pokazane okno dialogowe *Rejestrowanie makr* pochodzi z programu Word



RYSUNEK 1.4. Do pracy z makrami można wykorzystywać zakładkę *Deweloper* na *Wstążce*

Oto opis podstawowych funkcji dotyczących Visual Basica, dostępnych na zakładce *Deweloper* na Wstążce (lub na zakładce *Narzędzia bazy danych* programu Access):

Przycisk *Uruchom makro*. Ten przycisk znajduje się tylko na Wstążce programu Access. Jego kliknięcie powoduje wyświetlenie okna dialogowego *Uruchom makro*, w którym można wybrać makro do uruchomienia. Wiele elementów obsługi VBA w programie Access jest unikatowych, przeznaczonych wyłącznie dla Accessa. Szczegółowo omówiono je w rozdziale 28., „Model obiektów Accessa i najważniejsze obiekty”.

Przycisk *Zarejestruj makro*. Wyświetla okno dialogowe *Rejestrowanie makra* w programach Word lub Excel.

Przycisk *Bezpieczeństwo makr*. Wyświetla okno dialogowe ustawień makra *Centrum zaufania*. Ta własność zostanie szczegółowo omówiona w rozdziale 19. Przycisk ten pozwala określić, czy i w jaki sposób chcesz udostępnić makra.

Przycisk *Visual Basic*. Otwiera okno programu Visual Basic Editor. Pracę z tym programem rozpoczniemy w rozdziale 2., „Wprowadzenie do programu Visual Basic Editor” (a podczas lektury większości pozostałej części książki będziesz intensywnie z niego korzystał).

Przycisk *Makra*. Otwiera klasyczne okno dialogowe makr, z poziomu którego można uruchomić makro, wykonać je krok po kroku (uruchomić edytor Visual Basica w trybie *Break* — więcej informacji na ten temat znajdziesz w rozdziale 3., „Edycja zarejestrowanych makr”), edytować je, tworzyć, usuwać lub otworzyć okno dialogowe organizatora projektu makra (nie wszystkie wymienione opcje są dostępne we wszystkich aplikacjach; na przykład w PowerPoincie nie ma organizatora). W Wordzie i Excelu podobny przycisk *Makra* znajduje się na zakładce *Widok* na Wstążce. Ten przycisk pozwala na otwarcie okna dialogowego *Makra*, ale za jego pomocą można również rozpocząć nagrywanie makra. Należy zapamiętać, że tryb *Break* jest również określany jako tryb krokowy.

Dodatki. Z tego poziomu można uzyskać dostęp do szablonów, stylów i specjalistycznych bibliotek kodu.

Formanty. Zbiór przycisków sterujących, które po kliknięciu powodują wstawienie do otwartego dokumentu komponentów interfejsu użytkownika, takich jak rozwijane listy. Podobne komponenty mogą być również dodawane do makr tworzonych w edytorze VBA. Zagadnienia tworzenia elementów interfejsu użytkownika omówimy w rozdziałach 14., „Tworzenie prostych niestandardowych okien dialogowych”, i 15., zatytułowanym „Tworzenie złożonych formularzy”.

Przycisk *Tryb projektowania*. Przełącza pomiędzy trybem projektowania a trybem normalnym. W trybie projektowania można dodawać lub edytować osadzone kontrolki dokumentów. W trybie normalnym można komunikować się z kontrolkami (zwykle kontrolki pozwalają na pobieranie od użytkownika informacji za pośrednictwem wpisywania lub kliknięć myszą).

Przycisk *Właściwości*. Ten przycisk jest aktywny tylko w trybie projektowania. Pozwala edytować właściwości dokumentu (na przykład usuwać dane osobowe).

Przycisk *Mapowanie*. Tę sekcję zakładki *Deweloper* opisano w rozdziałach od 21. do 24.

Przycisk *Ogranicz edytowanie*. Pozwala określić, jakie operacje formatowania i edycji mogą wykonywać użytkownicy.

Przycisk *Szablony*. Pozwala zobaczyć lub zmodyfikować bieżący szablon, lub zarządzać dodatkami bądź szablonem globalnym.

POJAWIENIE SIĘ XML

XML stał się standardem branżowym do przechowywania i przenoszenia danych. Począwszy od wersji Office 2007, w dokumentach aplikacji Office zaczęto powszechnie stosować XML. Przejście na XML jest głównym powodem, dla którego dokumenty utworzone w wersjach pakietu Office 2007, 2010, 2013 i 2019 nie są kompatybilne z tymi z wcześniejszych wersji, takimi jak dokumenty Office 2003. Z tego względu należy dokonać konwersji starszych dokumentów Office do nowszych formatów pakietu. Należy zwrócić uwagę, że począwszy od Worda 2010, pliki dokumentów są zapisywane z rozszerzeniem *.docx*. Litera *x* nawiązuje do stosowania formatu XML, na którym bazuje Office.

Nazewnictwo makr

Po uruchomieniu Rejestratora należy wprowadzić nazwę dla nowego makra w polu *Nazwa makra* w oknie dialogowym *Rejestrowanie makr*. Nazwa musi spełniać następujące konwencje:

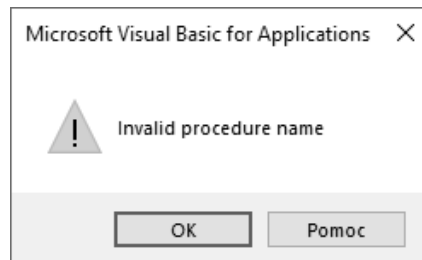
- musi zaczynać się od litery; za literą może zawierać zarówno litery, jak i cyfry;
- może mieć długość do 80 znaków;
- może zawierać podkreślenia, które są przydatne do oddzielania słów, na przykład `file_save`;
- nie może zawierać spacji, znaków interpunkcyjnych lub specjalnych, takich jak ! lub *.

Nieprawidłowe nazwy makr

Word i Excel nie pozwalają na wprowadzenie niepoprawnych nazw makra. Jeśli wpiszesz niedozwoloną nazwę makra w oknie dialogowym *Rejestrowanie makr*, aplikacje te poinformują Cię o tym na swój sposób, jak tylko klikniesz przycisk *OK*. Word oraz Excel wyświetlą krótki komunikat. Na rysunku 1.5 pokazano, jak wymienione aplikacje reagują na wprowadzenie niepoprawnej nazwy makra.

RYSUNEK 1.5.

Okno dialogowe wyświetlane przez programy Word i Excel w przypadku wprowadzenia nieprawidłowej nazwy makra



Opisywanie makr

Opis makra należy wpisać w polu tekstowym *Opis*. Warto pamiętać, że ma on pomóc nam (i wszystkim, którym udostępniamy makro) zidentyfikować makro i zrozumieć, kiedy należy go używać. Jeśli makro działa skutecznie tylko w określonych warunkach, można je wymienić w polu tekstowym *Opis*. Na przykład, jeśli użytkownik musi zaznaczyć tekst w dokumencie przed uruchomieniem makra w programie Word, można o tym wspomnieć.

NAZYWANIE I OPISYWANIE MAKR

Niektórzy ludzie twierdzą, że aby właściwie zarządzać zestawem makr, trzeba przestrzegać pewnych procedur związanych z nadawaniem im opisowych nazw, a także wpisywaniem narracyjnego opisu dla każdego celu spełnianego przez makro. Twierdzą oni, że jeśli utworzymy wiele makr, powinniśmy je starannie organizować. Nagrywanie makr jest bardzo proste, dzięki czemu można tworzyć kod bardzo łatwo i szybko. W efekcie może dojść do powstania sterty makr. W takim „gąszczu” łatwo się zagubić i trudno stwierdzić, które makro co robi.

Często, kiedy się śpieszymy lub gdy szukamy różnych sposobów podejścia do problemu, może nas kusić, aby nie przypisywać makrom opisów. W takiej sytuacji często nie jesteśmy pewni, które z testowych makr zatrzymać (jeśli w ogóle którekolwiek).

W przypadku prostego, oczywistego kodu, być może użycie nazw domyślnych (takich jak `Macro12`, `Macro13`) i pominięcie opisu nie stanowi problemu. Ponadto po uzyskaniu wprawy w czytaniu i rozumieniu kodu VBA, zwykle wystarczy spojrzeć na makro, by zorientować się, co ono robi.

Pomimo to, w przypadku bardziej złożonych makr warto wprowadzić kilka notatek dla każdego makra, które rejestrujesz, zwłaszcza jeśli czytanie kodu sprawia Ci trudności. W przeciwnym razie może dojść do powstania stosu zarejestrowanych makr, które mają tajemnicze nazwy domyślne i żadnych opisów. Aby dowiedzieć się, co robią makra, z których już nie korzystasz i mógłbyś je bezpiecznie usunąć, konieczne będzie przejrzanie całego kodu — a trzeba pamiętać, że kod zarejestrowanych makr może być zaskakująco długi nawet wtedy, gdy makro nie robi niczego więcej niż ustawienie kilku opcji w kilku oknach dialogowych.

Warto również stosować konwencję nazewnictwa makr, za pomocą której można wskazać makra testowe, które można bezpiecznie usunąć. Wystarczy rozpocząć nazwę od słowa takiego jak *Test*, a następnie dodawać kolejne wartości liczbowe, oznaczające wersje — na przykład *Próba* (*Próba01*, *Próba02* i tak dalej) lub *Temp* (*Temp01*, *Temp02* i tak dalej).

Każde nowe nagrywane makro jest domyślnie umieszczane w edytorze VBA w dolnej części zbioru makr. Zawsze jednak możesz otworzyć program Visual Basic Editor i zmienić nazwę lub dodać opis, ponieważ makra są w pełni edytowalne.

Osobiście jestem zwolennikiem umieszczania opisowej uwagi na początku kodu bardziej skomplikowanych makr, bezpośrednio pod wierszem ze słowem kluczowym `Sub`. Wygląda to następująco:

```
Sub AltH()  
  'Stosuje styl Nagłówek 1  
  Selection.Style = ActiveDocument.Styles("Nagłówek 1")  
End Sub
```

Dowolny tekst występujący za symbolem apostrofu (') w wierszu kodu jest ignorowany przez VBA. Pojedynczy apostrof wskazuje, że to, co za nim występuje, jest *komentarzem*, który ma pomóc programiście w zrozumieniu kodu, a nie rzeczywistym kodem VBA do wykonania (VBA nie będzie wiedział, co zrobić ze słowami *Stosuje styl Nagłówek 1*; nie należą one do słownictwa języka VBA).

Zauważ, że jeśli wpiszesz opis w polu *Opis* okna dialogowego *Rejestrowanie makr*, gdy po raz pierwszy rozpoczniesz nagrywanie, ten komentarz zostanie automatycznie umieszczony w kodzie — razem z symbolem apostrofu.

Poza tym, moim ulubionym sposobem nazywania makr, które są wywoływane za pomocą skrótów klawiaturowych, jest użycie nazwy samego skrótu klawiaturowego. Zatem `Sub AltH` mówi mi, że to makro jest wyzwalane przez skrót klawiszowy *Alt+H*.

Jednak niezależnie od stosowanego systemu, na ogół lepiej jest, gdy w kodzie jest za dużo opisów bądź komentarzy niż wtedy, gdy jest ich za mało. Dodanie opisowej nazwy i czytelnego opisu celu makra zajmuje tylko chwilę.

Należy także zdecydować, gdzie przechowywać makro. Dostępne wybory dla Worda i Excela są następujące:

Word. Przypomnijmy, że jeśli w programie Word chcemy ograniczyć dostępność makra do bieżącego szablonu (plik *.dotm*) lub dokumentu (plik *.docm*), powinniśmy wybrać ten szablon lub dokument z listy rozwijanej *Przechowaj makro w*, dostępnej w oknie dialogowym *Rejestrowanie makra* pokazanym na rysunku 1.3. Jeśli chcesz, aby makro było dostępne niezależnie od szablonu, z którym pracujesz, upewnij się, że w polu kombi *Przechowaj makro w* znajduje się domyślne ustawienie — *Wszystkie dokumenty (Normal.dotm)* (jeśli nie jest dla Ciebie jasne, czym są szablony Worda i co robią, zapoznaj się z treścią ramki „Normal.dotm, szablony i dokumenty w Wordzie” w dalszej części tego rozdziału).

Excel. W programie Excel są dostępne następujące lokalizacje do zapisania makra: *Ten skoroszyt*, *Nowy skoroszyt* lub *Skoroszyt makr osobistych*. Skoroszyt makr osobistych jest specjalnym skoroszytem o nazwie *Personal.xlsb*. Excel tworzy go w chwili, gdy po raz pierwszy wybierzemy opcję *Skoroszyt makr osobistych* jako lokalizację przechowywania makra. Dzięki utrzymywaniu makr i innych spersonalizowanych ustawień w skoroszycie makr osobistych, mogą być one dostępne we wszystkich tworzonych procedurach. Przypomnijmy, że skoroszyt makr osobistych przypomina globalny plik przechowywania makr Worda — *Normal.dotm*. Jeśli wybierzesz opcję *Nowy skoroszyt*, Excel utworzy dla Ciebie nowy skoroszyt i będzie zapisywać w nim makra.

Przechowywanie makr

Word i Excel automatycznie zapisują zarejestrowane makra w domyślnej lokalizacji w określonym dokumencie, szablonie, skoroszycie lub prezentacji.

Word. Word zapisuje każde zarejestrowane makro w *module* o nazwie *NewMacros* w wybranym szablonie lub dokumencie. Dzięki temu po zarejestrowaniu makra zawsze będziesz wiedzieć, gdzie je znaleźć. Może to być nieco mylące, ponieważ może być wiele folderów *NewMacros* widocznych w panelu Project Explorer w programie Visual Basic Editor (dzieje się tak dlatego, że może być otwarty więcej niż jeden projekt — na przykład kilka jednocześnie otwartych dokumentów, każdy z własnym folderem *NewMacros*). Pomyśl o folderze *NewMacros* jedynie jako o tymczasowym obszarze przechowywania makr — będą tam do czasu, aż przeniesiesz je do innego modułu o bardziej opisowej nazwie (oczywiście, jeśli stworzysz tylko kilka makr, nie musisz zadawać sobie trudu tworzenia różnych specjalnych modułów, aby podzielić makra na kategorie; możesz pozostawić je wszystkie w module *NewMacros* — jak zawsze wszystko zależy od tego, jak dobrze mamy zorganizowany umysł i pamięć, a także od wielkości zbioru makr, z którym mamy do czynienia).

Jeżeli moduł *NewMacros* jeszcze nie istnieje, to Rejestrator makr go stworzy. Ponieważ do modułu *NewMacros* trafia każde makro zarejestrowane w dokumencie lub szablonie, to jeśli rejestrujesz wiele makr, może on szybko rozrosnąć się do dużych rozmiarów. Moduł *NewMacros* w domyślnym globalnym szablonie *Normal.dotm* jest szczególnie narażony na szybkie rozrastanie się, ponieważ tutaj trafiają wszystkie zarejestrowane makra, o ile przed ich zarejestrowaniem nie podamy innego dokumentu lub szablonu. Niektórzy użytkownicy Worda od czasu do czasu czyszczą zawartość modułu *NewMacros*, przenosząc nagrane makra, które chcą zachować, do innych modułów, i usuwając wszystkie niepotrzebne lub tymczasowe makra. Osobiście nie mam tak wielu makr, więc po prostu nie mam nic przeciw temu, aby po prostu pozostały w module *NewMacros*.

Excel. Excel przechowuje wszystkie nagrane makra dla każdej sesji w nowym module o nazwie *Modulen*, gdzie *n* jest najniższą nieużywaną liczbą w kolejności rosnącej (*Module1*, *Module2* i tak dalej). Wszystkie makra stworzone w następnej sesji są przechowywane w nowym module, z kolejnym dostępnym numerem. A zatem, jeśli często rejestrujesz makra w Excelu, najprawdopodobniej będziesz musiał skonsolidować (skopiować i wkleić) makra, które chcesz zachować, tak aby nie były porzucane po wielu modułach.

NORMAL.DOTM, SZABLONY I DOKUMENTY W WORDZIE

Word, począwszy od wersji 2007, przechowuje dane inaczej niż we wcześniejszych wersjach. W programie Word 2003 można było tworzyć własne menu i paski narzędzi, które były przechowywane w szablonach. Późniejsze wersje programu Word nie pozwalają na definiowanie menu ani tworzenie jakichkolwiek innych pasków narzędzi niż pasek *Szybki dostęp*. Co więcej, personalizacja tego paska jest globalna. Innymi słowy, wszelkie modyfikacje wprowadzone na pasku narzędzi *Szybki dostęp* będą widoczne we wszystkich dokumentach Worda, bez względu na szablon (lub szablony), który jest aktualnie aktywny.

W Wordzie od wersji 2007 do 2019 występują trzy rodzaje szabloneń:

- ◆ Szablony tradycyjne z Worda 2003 i wersji wcześniejszych. Mają one rozszerzenie nazwy pliku *.dot*. Jeśli pracujesz z jednym z tych szabloneń, na pasku tytułu programu Word pojawia się wyrażenie (*tryb zgodności*).
- ◆ Szablony, które nie zawierają makr, są zapisane w pliku z rozszerzeniem *.dotx*. Można zapisywać makra w dokumencie, który wykorzystuje szablon *.dotx*, ale makra nie będą zapisane wewnątrz szablonu.
- ◆ Szablony z rozszerzeniem *.dotm* zawierają makra. Przypomnijmy, że z powodu tego, że makra napisane przez złośliwych użytkowników mogą, podobnie jak wirusy, wyrządzać szkody, w Wordzie umieszczono makra w tych szczególnego rodzaju szablonach z rozszerzeniem *.dotm*. Szablony *.dotm* mogą robić wszystko to, co umożliwia szablon *.dotx*, ale szablon *.dotm* jest wyposażony w dodatkowe możliwości hostingu makr.

Word ma architekturę czterowarstwową. Zaczynając od dołu, te warstwy to: sama aplikacja, szablon globalny (*Normal.dotm*), aktywny szablon dokumentu i wreszcie sam aktywny dokument (tekst i formatowanie). Każda z czterech warstw może wpływać na wygląd dokumentu Worda i jego zachowanie, ale niekoniecznie wszystkie cztery są w danym momencie aktywne.

Dolna warstwa, która jest zawsze aktywna, to sama aplikacja Worda. Warstwa ta zawiera wszystkie obiekty Worda i wbudowane polecenia, takie jak *Otwórz*. Również zawsze aktywne są takie obiekty jak pasek *Szybki dostęp* Worda, *Wstążka* i tym podobne. Ta warstwa jest najtrudniejsza do zobrazowania, ponieważ zwykle nie widzimy jej bezpośrednio. *Normal.dotm*, szablon globalny, tworzy drugą warstwę i także jest zawsze aktywny.

Po uruchomieniu program Word ładuje szablon *Normal.dotm* automatycznie i pozostaje on załadowany do chwili zamknięcia Worda (istnieje specjalny przełącznik — *winword/n* — którego użycie pozwala zapobiec aktywności makr w szablonie *Normal.dotm*, jeśli trzeba rozwiązywać związane z nimi problemy; aby uruchomić program Word w ten szczególny sposób, naciśnij klawisz *Start* — *Windows* w *Windows 8* i *10*, *Start* we wcześniejszych wersjach systemu — a następnie wpisz polecenie *winword/n* w oknie *Uruchom*).

Szablon *Normal.dotm* zawiera style (takie jak domyślny styl akapitu), wpisy Autotekstu, sformatowane wpisy autokorekty i ustawienia spersonalizowane. Te personalizacje są widoczne również w innych warstwach, o ile nie zostaną wyraźnie wykluczone.

Domyślnie nowe puste dokumenty (takie jak dokument, który Word tworzy zazwyczaj po jego uruchomieniu, oraz dowolne inne dokumenty, które stworzymy po kliknięciu *CTRL+N* lub kliknięciu zakładki *Plik* na *Wstążce*, a następnie wybraniu Polecenia *Nowy/Pusty dokument*) są oparte na szablonie *Normal.dotm*. W związku z tym, kiedy pracujesz z domyślnym, pustym dokumentem, komunikujesz się z interfejsem programu Word w takiej postaci, w jakiej zdefiniowano go w szablonie *Normal.dotm*.

Bieżący aktywny szablon działa w warstwie nad aplikacją Word i szablonem *Normal.dotm*. Szablon ten może zawierać style, moduły makr (jeśli jest to szablon w pliku *.dotm*) oraz ustawienia szablonu, wraz z dowolnym gotowym tekstem wymaganym dla danego typu dokumentu. Jest to trzecia warstwa, ale jest używana tylko wtedy, gdy dokument bieżący (lub aktywny) jest dowiązany do szablonu innego niż *Normal.dotm*.

W warstwie nad bieżącym szablonem działa bieżący dokument, który zawiera tekst i grafikę w dokumencie, formatowanie oraz układ. Dokumenty mogą również zawierać moduły makr specyficznych dla danego dokumentu wraz ze spersonalizowanymi skrótami klawiaturowymi, więc sam dokument może pełnić rolę czwartej warstwy. Warstwa ta występuje zawsze, gdy dokument jest otwarty, ale nie ma wpływu na interfejs lub zachowania Worda, chyba że dokument zawiera własne, lokalne personalizacje.

Ponieważ warstwy te mogą zawierać sprzeczne informacje (np. dwa różne style czcionek o tej samej nazwie), musi istnieć porządek pierwszeństwa pozwalający określić, która warstwa „wygrywa” w takim konflikcie (jest to podobne do *kaskady* w programowaniu CSS). Spersonalizowane ustawienia działają począwszy od górnej warstwy w dół. W związku z tym niestandardowe ustawienia w aktywnym dokumencie mają pierwszeństwo nad tymi w aktywnym szablonie. Podobnie wszelkie ustawienia w aktywnym szablonie mają pierwszeństwo przed wszelkimi szablonami globalnymi (czyli tymi, które są stosowane automatycznie do wszystkich dokumentów Worda) lub dodatkami innymi niż *Normal.dotm*. Spersonalizowane ustawienia w tych szablonach globalnych lub dodatkach mają pierwszeństwo nad tymi, które są zdefiniowane w szablonie *Normal.dotm*.

Oto inny przykład. Powiedzmy, że posługujemy się kombinacją klawiszy *Ctrl+Shift+K*, którą przypisano do różnych działań w szablonie *Normal.dotm*, w załadowanym szablonie globalnym, w szablonie dokumentu oraz w samym dokumencie. Po naciśnięciu tej kombinacji klawiszy, zostanie uruchomiona tylko procedura przypisana w dokumencie, ponieważ to jest najwyższa warstwa. Jeśli usuniesz przypisanie kombinacji klawiszy z dokumentu, najwyższą warstwą zawierającą definicję tej kombinacji klawiszy stanie się szablon, więc uruchomi się procedura przypisana w szablonie. Jeśli usuniesz kombinację klawiszy także z szablonu, uruchomi się procedura w załadowanym szablonie globalnym. Wreszcie, jeśli usuniesz tę kombinację klawiszy również z szablonu globalnego, uruchomi się procedura z szablonu *Normal.dotm*. Jest to najniższa warstwa.

Wybór sposobu uruchomienia nowego makra

Kontynuujemy eksplorację okna dialogowego *Rejestrowanie makr* pokazanego na rysunku 1.3. W tym momencie, po nadaniu makru nazwy, wpisaniu opisu oraz wyborze, gdzie należy je przechowywać, nadszedł czas, aby zdecydować o sposobie wywoływania makra. Innymi słowy, trzeba określić, w jaki sposób użytkownik ma je uruchamiać: za pomocą klawisza skrótu, czy za pomocą przycisku na pasku narzędzi *Szybki dostęp*? Osoby sprawnie posługujące się klawiaturą zazwyczaj preferują klawisze skrótów (nie muszą odsuwać ręki od klawiatury, aby sięgnąć po mysz). Jednak przyciski zapewniają wizualną wskazówkę celu makra. Ponadto naprowadzenie wskaźnika myszy na przycisk wyświetla również nazwę makra.

Skróty klawiszowe i przyciski są przydatne dla osób, które rejestrują umiarkowaną liczbę makr i nie organizują ich w bardzo złożony sposób, poprzez przenoszenie z jednego modułu do drugiego. Jeśli utworzysz dużą liczbę makr i czujesz potrzebę, aby przenieść je do innych modułów, przypisanie klawisza skrótu lub przycisku przed rejestrowaniem staje się mniej przydatne. To dlatego, że przemieszczanie makra z jednego modułu do innego ukrywa sposób przypisany do uruchamiania makra.

Ograniczenie to oznacza, że przypisywanie sposobu uruchamiania makra przed jego zarejestrowaniem ma sens tylko wtedy, gdy planujesz używać makra w jego zarejestrowanej postaci (w przeciwieństwie do, powiedzmy, wykorzystania jego części w celu stworzenia kolejnego makra) oraz w zależności od lokalizacji modułu. Jeśli planujesz przenoszenie makra lub zmianę jego nazwy, nie przypisuj sposobu uruchamiania makra od razu. Zamiast tego poczekaj, aż makro będzie w swojej ostatecznej formie i lokalizacji, a następnie przypisz sposób jego uruchamiania. Szczegółowe informacje na temat tego, jak to zrobić, można znaleźć w punkcie „Określanie sposobu uruchamiania istniejącego makra” w dalszej części tego rozdziału.

Osobiście mam około kilkadziesiąt makr, których używam przez cały czas, więc unikam komplikacji opisanych w poprzednim akapicie oraz w ramce zatytułowanej „Zarządzanie makrami za pomocą modułów”. Zamiast tego, po prostu dodaję klawisz skrótu, kiedy tworzę makro po raz pierwszy, i pozostawiam wszystkie makra w jednej wersji w szablonie *Normal.dotm*. Jednakże w bardziej skomplikowanej sytuacji — na przykład w przypadku zarządzania dużym zbiorem makr w firmie — można zdecydować o zorganizowaniu makr za pomocą modułów.

ZARZĄDZANIE MAKRAMI ZA POMOCĄ MODUŁÓW

Aby przypisać sposób uruchamiania makra, postępuj zgodnie z instrukcjami w następujących punktach.

Pamiętaj, że nie trzeba przypisać przycisku lub skrótów klawiaturowego przed zarejestrowaniem makra. Można to zrobić później, w dowolnym momencie. W programie Word i Excel, aby przypisać do makra przycisk na pasku narzędzi *Szybki dostęp*, można użyć polecenia *Plik/Opcje/Pasek narzędzi Szybki dostęp*. Można też, by przypisać klawisz skrótów do makra, użyć polecenia *Plik/Opcje/Dostosuj Wstążkę* (kiedy otworzy się okno dialogowe, kliknij przycisk *Dostosuj* u dołu tego okna).

W Excelu obowiązuje ograniczenie do stosowania kombinacji klawiszy *Ctrl+* lub *Ctrl+Shift*. W PowerPointie i Accessie nie można przypisywać skrótów klawiszowych do makr.

Uruchamianie makra ze Wstążki

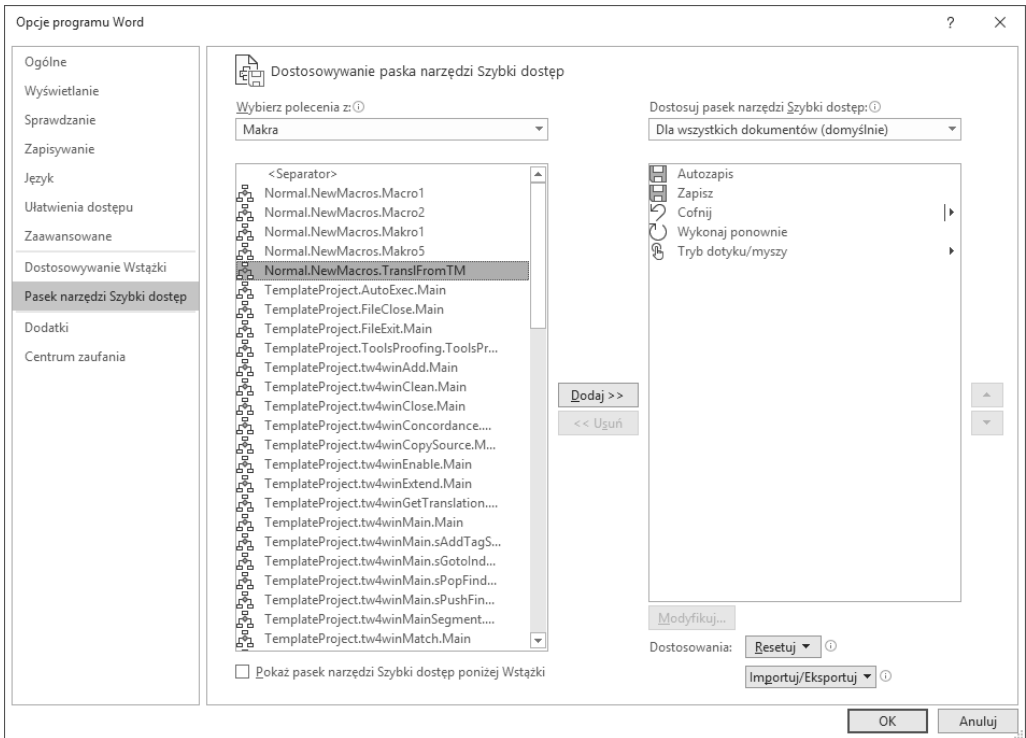
Chociaż nie jest to dostępne w oknie dialogowym *Rejestrowania makra*, można dodać makro do Wstążki. Aby to zrobić, wykonaj następujące czynności:

1. Kliknij prawym przyciskiem myszy w dowolnym miejscu Wstążki.
2. Kliknij przycisk *Dostosuj Wstążkę* w menu kontekstowym, które się wyświetli. Otworzy się okno dialogowe *Opcje programu Word*.
3. Na rozwijanej liście *Wybierz polecenia* z wskaż pozycję *Makra*.
4. Kliknij nazwę makra, aby je zaznaczyć na liście.
5. Kliknij istniejącą zakładkę na liście zakładek w oknie dialogowym po prawej, wskazując tym samym, gdzie chcesz umieścić makro.
6. Następnie kliknij przycisk *Nowa grupa* i podaj nazwę niestandardowej grupy.
7. Kliknij przycisk *Zmień nazwę*, aby nadać grupie nową nazwę.
8. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Zmianianie nazwy*.
9. Kliknij przycisk *Dodaj*, aby dodać swoje makro.
10. Kliknij przycisk *Zmień nazwę*, aby nadać makru zrozumiałą nazwę i ewentualnie ikonę.
11. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Zmianianie nazwy*.
12. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Opcje programu Word*.

Uruchamianie makra z paska narzędzi Szybki dostęp

Oto jak można użyć okna dialogowego *Opcje programu Word*, aby przypisać makro do przycisku na pasku narzędzi *Szybki dostęp*:

1. Kliknij prawym przyciskiem myszy w dowolnym miejscu na pasku narzędzi *Szybki dostęp* (jest to zbiór ikon w lewym górnym rogu ekranu, zwykle powyżej Wstążki). Wyświetli się menu.
Ten pasek narzędzi znajduje się bezpośrednio pod Wstążką, jeśli wcześniej wybrałeś z tego menu opcję *Pokaż pasek narzędzi Szybki dostęp poniżej Wstążki*.
2. Kliknij opcję *Dostosuj pasek narzędzi Szybki dostęp*.
Wyświetli się okno dialogowe *Opcje programu Word*.
3. W polu listy rozwijanej *Wybierz polecenia* z wskaż pozycję *Makra*.
4. Kliknij nazwę makra, aby je zaznaczyć na liście, tak jak pokazano na rysunku 1.6.



RYSUNEK 1.6. Wybierz sposób uruchamiania makra w oknie dialogowym Opcje programu Word

5. Kliknij przycisk *Dodaj*, aby wstawić nazwę tego makra na listę *Dostosuj pasek narzędzi Szybki dostęp*, jak pokazano na rysunku 1.6.

Word doda dla wskazanego makra przycisk na pasku narzędzi, nadając mu pełną kwalifikowaną nazwę (tworzy ją lokalizacja makra wraz z jego nazwą), taką jak `Normal.NewMacros.CreateDailyReport`. Nazwa ta składa się — odpowiednio — z nazwy szablonu lub dokumentu, w którym przechowywane jest makro, nazwy modułu, który zawiera makro, oraz nazwy makra. Nie trzeba wyświetlać wszystkich tych informacji po naprowadzeniu kursora myszy nad przycisk.

6. Aby zmienić nazwę przycisku pozycji w menu, kliknij przycisk *Modyfikuj* w dolnej części listy *Dostosuj pasek narzędzi Szybki dostęp*.

Modyfikowane będzie to makro, które jest podświetlone (aktualnie wybrane) na liście elementów paska narzędzi.

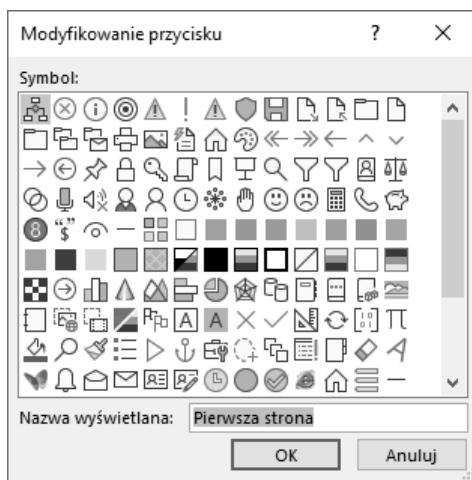
ETYKIETY PRZYCSKÓW MAKR NIE MUSZĄ BYĆ ZGODNE Z ICH OFICJALNYMI NAZWAMI

Zauważ, że nazwa przycisku makra (wyświetlana jako podpis podpowiedzi po naprowadzeniu kursora myszy nad przycisk) nie musi mieć żadnego związku z rzeczywistą nazwą makra, występującą w programie *Visual Basic Editor* lub oknie dialogowym *Makra*.

- Podczas modyfikowania nazwy makra możesz również wybrać ikonę przycisku, która optycznie wskazuje na przeznaczenie makra (patrz rysunek 1.7). Aby to zrobić, po prostu kliknij dwukrotnie ikonę, której chcesz używać, a następnie kliknij przycisk OK.

RYСУNEK 1.7.

Word nadaje pozycji menu lub przyciskowi paska narzędzi pełną nazwę makra. Użyj okna Modyfikowanie przycisku, aby zmienić nazwę na krótszą i lepszą



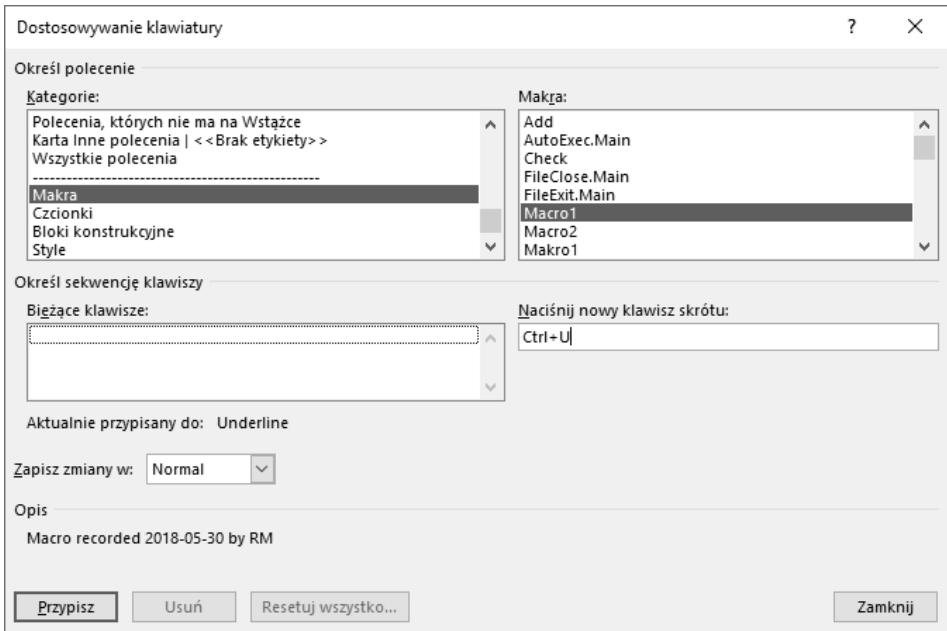
Uruchamianie makra za pomocą kombinacji klawiszy skrótu

Aby przypisać makro do kombinacji klawiszy, wykonaj następujące kroki:

- Kliknij prawym przyciskiem myszy Wstążkę i z menu, które się wyświetli, wybierz polecenie *Dostosuj Wstążkę*.
Otworzy się okno dialogowe *Opcje programu Word*.
- Kliknij przycisk *Dostosuj* obok przycisku *Skróty klawiaturowe* w lewym dolnym rogu okna dialogowego *Opcje programu Word*.
- Przewiń w dół listę *Kategorie*, aż zobaczysz pozycję *Makra*, a następnie kliknij tę pozycję, aby ją wybrać.
- Kliknij nazwę makra, do którego chcesz przypisać skrót w postaci kombinacji klawiszy.
- Sprawdź listę *Bieżące klawisze*, aby upewnić się, czy kombinacja klawiszy nie jest już przypisana.

Jeśli tak jest, możesz nacisnąć klawisz *Backspace*, aby usunąć tę kombinację klawiszy. Możesz też wykorzystać wiele kombinacji klawiszy do uruchomienia makra.

- W polu *Naciśnij nowy klawisz skrótu* wprowadź kombinację klawiszy, której chcesz użyć do uruchomienia makra (patrz rysunek 1.8).
- Sprawdź, czy ta kombinacja klawiszy nie jest już używana w innym celu.
Jeśli tak jest, to możesz zmienić jej przypisanie lub możesz wybrać inną kombinację, naciskając klawisz *Backspace* w polu *Naciśnij nowy klawisz skrótu*.
- Pamiętaj, aby kliknąć przycisk *Przypisz*, kiedy wszystko będzie gotowe.
Same zamknięcie tego okna nie spowoduje przypisania kombinacji klawiszy.



RYSUNEK 1.8. Ustaw kombinację klawiszy dla makra w oknie dialogowym Dostosuj klawiaturę

MOŻESZ ODŁOŻYĆ NA PÓŹNIEJ PRZYPISANIE KOMBINACJI KLAWISZY SKRÓTU

Pamiętaj, że tak samo jak w przypadku różnych sposobów uruchamiania makra, kombinację klawiszy skrótów można przypisać zarówno w czasie rejestrowania makra, jak i w dowolnym momencie po zakończeniu rejestrowania. Jeśli zamierzasz przenieść makro z modułu *NewMacros* do innego, pamiętaj, że nie musisz przypisywać kombinacji klawiszy uruchamiających makro przed tym, zanim dotrze ono do swojego ostatecznego miejsca przeznaczenia.

Kombinacje klawiszy w programie Word mogą mieć jedną z następujących form:

- *Alt* plus klawisz funkcyjny albo zwykły klawisz, który nie jest używany jako klawisz dostępu do menu.
- *Ctrl* plus klawisz funkcyjny lub zwykły klawisz.
- *Shift* plus klawisz funkcyjny.
- *Ctrl+Alt*, *Ctrl+Shift*, *Alt+Shift* lub nawet *Ctrl+Alt+Shift* plus zwykły klawisz lub klawisz funkcyjny. Wcisnięcie *Ctrl+Alt+Shift* wraz z innym klawiszem wydaje się być zbyt niewygodne do praktycznego zastosowania.

OKREŚL DWUETAPOWĄ KOMBINACJĘ KLAWISZY

Można skonfigurować klawisze skrótów, które składają się z dwóch etapów, na przykład *Ctrl+Alt+F, 1* lub *Ctrl+Alt+F, 2*. Drugi klawisz (w tym przypadku *1* lub *2*) należy nacisnąć po naciśnięciu kombinacji klawiszy. Takie skróty sprawiają jednak zbyt dużo kłopotów, więc nie warto ich stosować.

Uruchamianie makra w staromodny sposób

Niezdarny, rzadko używany sposób na uruchomienie makra polega na kliknięciu zakładki *Deweloper* na Wstążce. Aby zobaczyć, jak to działa, wykonaj następujące kroki:

1. Kliknij ikonę *Makra*.
2. Na liście, która się wyświetli, kliknij nazwę makra.
3. Na koniec kliknij przycisk *Uruchom*.

Nawiasem mówiąc, można także uruchomić makro z poziomu programu Visual Basic Editor. W tym celu należy nacisnąć klawisz *F5*. W ten sposób można przetestować makro podczas jego edycji. Po naciśnięciu klawisza *F5* w edytorze uruchomi się makro, w którym znajduje się kursor wstawiania (migająca pionowa linia).

Przypisywanie sposobu uruchamiania makra w programie Excel

Podczas rejestrowania makra Excel pozwala na przypisanie tylko klawiszy skrótu z *Ctrl*. Nie pozwala na stworzenie przycisku, który uruchamia makro. Aby przypisać do makra przycisk na pasku narzędzi *Szybki dostęp*, trzeba to zrobić po zakończeniu rejestrowania makra (za pomocą funkcji *Dostosuj*, którą wkrótce opiszemy).

Aby przypisać klawisz skrótu *Ctrl* uruchamiający makro, które zarejestrowałeś, wykonaj następujące kroki:

1. Rozpocznij rejestrowanie makra, aby wyświetlić okno dialogowe *Rejestrowanie makra*, a następnie kliknij pole tekstowe *Klawisz skrótu Ctrl+*, tak aby znalazł się w nim migający kursor wstawiania.
2. Naciśnij klawisz skrótu, którego chcesz użyć (naciśnij jednocześnie klawisz *Shift*, jeśli chcesz użyć tego klawisza w skrócie).
3. Na rozwijanej liście *Przechowuj makro w* określ, gdzie chcesz, aby Rejestrator makr przechowywał makro. Masz do wyboru następujące opcje:
 - ◆ *Ten skoroszyt* — przechowuje makra w aktywnym skoroszytcie. Ta opcja jest przydatna dla makr, które należą do określonego skoroszytu i nie muszą być wykorzystywane gdzie indziej.
 - ◆ *Nowy skoroszyt* — Excel utworzy nowy skoroszyt i zapisze w nim makro. Ta opcja jest przydatna dla makr eksperymentalnych, które będziesz edytować, zanim zaczniesz z nimi pracować.
 - ◆ *Skoroszyt makr osobistych* — makro zostanie zapisane w skoroszytcie makr osobistych, tzn. specjalnym skoroszytcie o nazwie *PERSONAL.XLSB*. Dzięki przechowywaniu makr oraz innych spersonalizowanych ustawień w skoroszytcie makr osobistych możesz je udostępnić we wszystkich tworzonych przez Ciebie procedurach. Stają się one globalne — dostępne we wszystkich skoroszytach. Jeśli skoroszyt makr osobistych jeszcze nie istnieje, Rejestrator makr stworzy go automatycznie, gdy użytkownik zdecyduje się zapisać makro z wykorzystaniem tej opcji.
4. Kliknij przycisk *OK*, aby rozpocząć nagrywanie makra.

Przypisywanie sposobu uruchamiania makra w programie PowerPoint

PowerPoint nie pozwala na rejestrowanie makr, ale można przypisać sposób uruchomienia makr napisanych w edytorze Visual Basic tak, jak opisano w punkcie „Określanie sposobu uruchamiania istniejącego makra” w dalszej części tego rozdziału.

Przypisywanie sposobu uruchamiania makra w programie Outlook

Outlook także nie pozwala na rejestrowanie makr, a ponadto domyślnie są one w nim wyłączone. Aby włączyć makra w programie Outlook, wykonaj następujące czynności:

1. Kliknij zakładkę *Deweloper* na Wstążce.
2. Kliknij ikonę *Bezpieczeństwo makr* (znajduje się po lewej stronie, w sekcji *Kod* na Wstążce).
Otworzy się okno dialogowe *Centrum zaufania*.
3. Kliknij opcję *Powiadomienia dla wszystkich makr* lub *Włącz wszystkie makra*.

Aby zobaczyć, jak przypisać sposób uruchomienia makra, zobacz punkt „Określanie sposobu uruchamiania istniejącego makra” w dalszej części tego rozdziału.

Rejestrowanie działań w makrze

Gdy zamkniesz okno dialogowe *Rejestrowanie makr* w programach Word lub Excel, Rejestrator makr rozpoczyna nagrywanie makra. Przypomnijmy, że Rejestrator makr wyświetla ikonę *Zatrzymaj rejestrowanie* (czarny kwadrat) na pasku stanu w lewym dolnym rogu ekranu (oraz przycisk *Zatrzymaj rejestrowanie* na zakładce *Deweloper* na Wstążce). Ponadto Word wyświetla niewielki symbol kasety na wskaźniku myszy (takie kasety były używane w dawnych czasach, przed wynalezieniem CD).

Teraz wykonaj sekwencję działań, które chcesz zarejestrować. To, co możesz zrobić, zależy od aplikacji, ale ogólnie rzecz biorąc, możesz użyć myszy do zaznaczenia elementów, dokonania wyboru w oknach dialogowych oraz wybierania zdefiniowane pozycje w dokumentach (na przykład komórki w arkuszach kalkulacyjnych). Niektórych działań za pomocą myszy nie można wykonać — na przykład nie da się zaznaczać elementów w oknie dokumentu w programie Word. Aby było to możliwe, trzeba skorzystać z klawiatury (na przykład wciskając *Shift+* klawisze strzałek).

REJESTRATOR MAKR REJESTRUJE WSZYSTKO — KOMPLETNY BIEŻĄCY STATUS

Kiedy dokonasz wyborów w oknie dialogowym i klikniesz przycisk *OK*, Rejestrator makr zapisze bieżące ustawienia wszystkich opcji dostępnych na tej stronie okna dialogowego. A zatem, na przykład, jeśli w oknie dialogowym *Akapit* w programie Word zmienisz wcięcie akapitu z lewej, Rejestrator makr zapisze także wszystkie inne ustawienia na stronie *Wcięcia i odstępy* (*Wyrównanie*, *Odstępy Przed i Po*, i tak dalej).

Jeśli w programie Word chcesz wykonać jakiegokolwiek czynności, których nie chcesz rejestrować, wstrzymaj działanie Rejestratora makr, klikając przycisk *Wstrzymaj rejestrowanie* na Wstążce. Etykieta przycisku zmienia się na *Wznów rejestrator*. Kliknij przycisk ponownie, aby ponownie rozpocząć rejestrowanie.

Aby zatrzymać nagrywanie, kliknij przycisk *Zatrzymaj rejestrowanie* na Wstążce lub drugi przycisk na pasku stanu.

Rejestrator makr właśnie zarejestrował makro i opcjonalnie przypisał je do kombinacji klawiszy skrótu lub do przycisku.

Uruchamianie makra

Aby uruchomić makro, które zarejestrowałeś, możesz użyć czterech metod:

- kliknięcie przycisku paska narzędzi *Szybki dostęp*, jeśli przypisałeś go do makra;
- kliknięcie przycisku na Wstążce, jeśli dodałeś do niej makro;

- wciśnięcie skrótu klawiaturowego, jeśli go przypisałeś;
- możesz też zastosować mniej wygodne rozwiązanie:
 - a. kliknij przycisk Deweloper/Makra, aby wyświetlić okno dialogowe Makra,
 - b. wybierz makro,
 - c. kliknij przycisk *Uruchom*.

Alternatywnie możesz kliknąć dwukrotnie nazwę makra w polu listy.

URUCHAMIANIE MAKRA W EDYTORZE VBA

Warto zapamiętać, że makro można uruchomić również z poziomu programu Visual Basic Editor. Jest to przydatne, gdy właśnie tworzysz lub modyfikujesz makro w edytorze. W celu uruchomienia makra wystarczy nacisnąć klawisz *F5*.

Gdy uruchomisz makro, to zostaną wykonane działania w takiej kolejności, w jakiej zostały nagrane. Załóżmy na przykład, że utworzyłeś w Excelu makro, które wybiera komórkę *A2* w bieżącym arkuszu, ustawia w niej pogrubioną czcionkę, wprowadza tekst *Roczna sprzedaż*, zaznacza komórkę *B2* i wpisuje w niej liczbę 100 000. Rejestrator makr rozpoznał i zapisał te pięć działań. Następnie VBA wykonuje je krok po kroku za każdym razem, gdy uruchomisz makro (choć robi to dość szybko).

JAK ZATRZYMAĆ DZIAŁAJĄCE MAKRO?

Aby zatrzymać działające makro, naciśnij kombinację klawiszy *Ctrl+Break* (*Break* na większości klawiatur to klawisz *Pause* wciskany bez *Shift*). VBA zatrzyma działanie programu i wyświetli okno dialogowe z informacją, że wykonywanie kodu zostało przerwane. Kliknij przycisk *End*, aby zamknąć to okno.

Niektóre aplikacje (takie jak Word) pozwalają cofnąć większość działań wykonywanych przez VBA, kiedy makro się zatrzyma (aby cofnąć pojedyncze polecenie, można nacisnąć *Ctrl+Z* lub kliknąć przycisk *Cofnij* na pasku narzędzi *Szybki dostęp*). Inne aplikacje na to nie pozwalają.

BŁĘDY MAKR SĄ CZĘSTO SPowodowane PRZEZ NIEPRAWIDŁOWY KONTEKST

Jeśli uruchomienie makra powoduje błąd, często oznacza to, że makro próbuje wykonać jakieś działanie na pliku lub innym obiekcie, który nie jest aktualnie dostępny. Na przykład, jeśli nagrasz w Excelu makro, które działa na aktywnym skoroszytcie, to makro to spowoduje błąd, jeśli spróbujesz je uruchomić bez otwarcia skoroszytu (co oznacza, że w tym momencie nie ma czegoś takiego jak *aktywny skoroszyt*). Podobnie, jeśli napiszesz makro wykonywane w programie PowerPoint, które współpracuje z trzecią ilustracją na aktywnym slajdzie, to makro to nie będzie działać, jeśli spróbujesz je uruchomić dla slajdu, na którym nie ma trzeciej ilustracji. Aby takie makra działały poprawnie, stwórz potrzebne warunki, a następnie ponów próbę.

Rejestrowanie przykładowego makra w programie Word

W tym punkcie nagramy przykładowe makro w programie Word. To makro zaznacza bieżące słowo, wycina je, przesuwa punkt wstawiania o jedno słowo w prawo i wkleja to samo słowo w nowym miejscu. Jest to prosta sekwencja działań, którą będziesz wykorzystywać w dalszej części książki, a także wyświetlać i edytować w programie Visual Basic Editor (dlatego nie usuwaj tego makra).

Aby zarejestrować makro, wykonaj następujące działania:

1. Utwórz nowy dokument, naciskając klawisze *Ctrl+N*.
2. Uruchom Rejestrator makr. Aby to zrobić, kliknij najpierw zakładkę *Deweloper* na Wstążce, a następnie przycisk *Zarejestruj makro*. Możesz też kliknąć przycisk *Zarejestruj makro* na pasku stanu u dołu aplikacji (jeśli zdecydujesz się na skorzystanie z tego skrótu, nie musisz otwierać zakładki *Deweloper*; wystarczy, że klikniesz przycisk na pasku stanu).
3. W polu tekstowym *Nazwa makra* wprowadź *Transpose_Word_Right*.
4. Upewnij się, że na rozwijanej liście *Przechowaj makro w* wybrano opcję *Wszystkie dokumenty (Normal.dotm)*, chyba że chcesz przypisać makro do innego szablonu.

W tym i w kolejnych przykładach w tej książce założono, że makra są przechowywane w szablonie *Normal.dotm*, więc tam będziemy je przechowywać.

5. W polu *Opis* wprowadź opis makra (patrz rysunek 1.8).

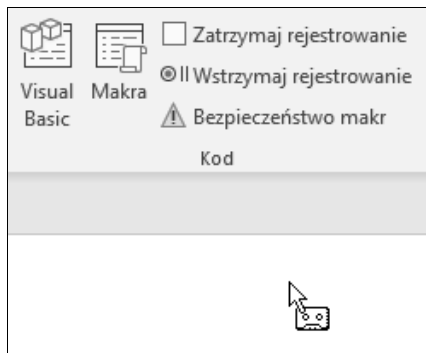
Zadbaj o to, by opis był czytelny. Może to być, na przykład, następujący opis: **Przenosi bieżące słowo za słowo występujące po jego prawej stronie. Utworzył Radek Meryk 28-09-2019.**

6. Jeśli chcesz, przypisz metodę uruchamiania makra, tak jak opisano nieco wcześniej. Utwórz przycisk paska narzędzi lub przypisz skrót klawiaturowy.

Wybór metody (lub metod) to wyłącznie kwestia osobistych preferencji. Jeśli będziesz musiał później przenieść makra do innego modułu (lub innego szablonu czy dokumentu), nie przypisuj metody uruchamianiu makra w tym momencie.

7. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Opcje programu Word* lub *Dostosuj klawiaturę* (lub po prostu kliknij przycisk *OK*, aby zamknąć okno dialogowe *Rejestrowanie makra*, jeśli zdecydowałeś się nie przypisywać sposobu uruchamiania makra).

Teraz jesteś gotowy do rejestrowania makra. Na Wstążce i na pasku stanu pojawi się opcja *Zatrzymaj rejestrowanie*, a wskaźnik myszy przyjmie postać ikony z wizerunkiem kasyety.



8. W ramach szybkiej demonstracji sposobu, w jaki możesz wstrzymać rejestrowanie, kliknij przycisk *Wstrzymaj rejestrowanie* na Wstążce.

Ikona przedstawiająca kasetę zniknie ze wskaźnika myszy, a etykieta na przycisku *Wstrzymaj rejestrowanie* zmieni się na *Wznów rejestrowanie*.

9. Wprowadź w dokumencie następujący wiersz tekstu: „**Szybki brązowy lis przeskoczył nad leniwym psem**”.
10. Umieść punkt wstawiania w dowolnym miejscu wyrazu „Szybki”, a następnie kliknij przycisk *Wznów rejestrowanie* na Wstążce, aby wznowić działanie Rejestratora makr.
11. Zarejestruj działania makra w następujący sposób:
 - a. użyj funkcji rozszerzania zaznaczenia Worda, aby wybrać słowo „Szybki”; w tym celu naciśnij dwukrotnie klawisz *F8*,
 - b. naciśnij klawisz *Esc*, aby anulować tryb rozszerzania,
 - c. naciśnij *Shift+Delete*, aby wyciąć zaznaczone słowo do schowka; punkt wstawiania jest teraz na początku słowa „brązowy”,
 - d. naciśnij *Ctrl+strzałka w prawo*, aby przenieść punkt wstawiania w prawo o jedno słowo; teraz będzie na początku słowa „lis”,
 - e. naciśnij *Shift+Insert* lub *Ctrl+V*, aby wkleić słowa wycięte ze schowka,
 - f. naciśnij *Ctrl+strzałka w lewo*, aby przenieść punkt wstawiania o jedno słowo w lewo; spowoduje to przywrócenie kursora do pozycji wyjściowej.
12. Kliknij przycisk *Zatrzymaj rejestrowanie* na Wstążce lub pasku stanu.
Twoje zdanie brzmi teraz: „brązowy Szybki lis przeskoczył nad leniwym psem”.

WYSZUKIWANIE WBUDOWANYCH SKRÓTÓW Klawiaturowych

Aby znaleźć pełną listę wbudowanych skrótów klawiaturowych (takich jak *Ctrl+strzałka w lewo*), można przeszukać system pomocy aplikacji w poszukiwaniu frazy „skrótów klawiaturowe” (w programie Office 2019 można po prostu użyć funkcji *Powiedz mi, co chcesz zrobić...* na Wstążce. Kliknij tę opcję, a następnie wpisz wyszukiwane hasło).

Teraz możesz uruchomić zarejestrowane makro, za pomocą przycisku paska narzędzi lub przypisanego skrótu klawiaturowego (o ile go wybrałeś). Możesz także kliknąć przycisk *Makra* na zakładce *Deweloper* i uruchomić makro z okna dialogowego *Makra*.

W tym momencie program Word zapisał makro w szablonie *Normal.dot*. Jeśli nie zapiszesz makra aż do momentu zamknięcia programu Word (lub do wykonania funkcji automatycznego tworzenia kopii zapasowej), Word domyślnie nie będzie Cię o to monitować. Po prostu zapisze je automatycznie. Najlepiej jednak kliknąć przycisk *Zapisz* na zakładce *Plik*, aby zapisać plik *Normal.dot*. Dzięki temu, jeśli Word lub Windows zawieszą się, unikniesz utraty makra.

MOŻESZ WYMUSIĆ, ABY WORD WYŚWIELAŁ MONIT O ZAPISANIE SZABLONU NORMAL

Word domyślnie automatycznie zapisuje nowe makra dodane do szablonu *Normal*. Jeśli jednak wolisz, aby wyświetlał monit przed zapisaniem jakichkolwiek zmian w tym szablonie, wybierz *Opcje* na zakładce *Plik* (przycisk *Opcje* został przesunięty do dolnej części ekranu), a następnie kliknij przycisk *Zaawansowane*. W kolejnym kroku przewiń ekran w dół, aż pojawi się sekcja opcji zapisu. Zaznacz pole wyboru *Monituj przed zapisaniem szablonu Normal*, po czym kliknij przycisk *OK*.

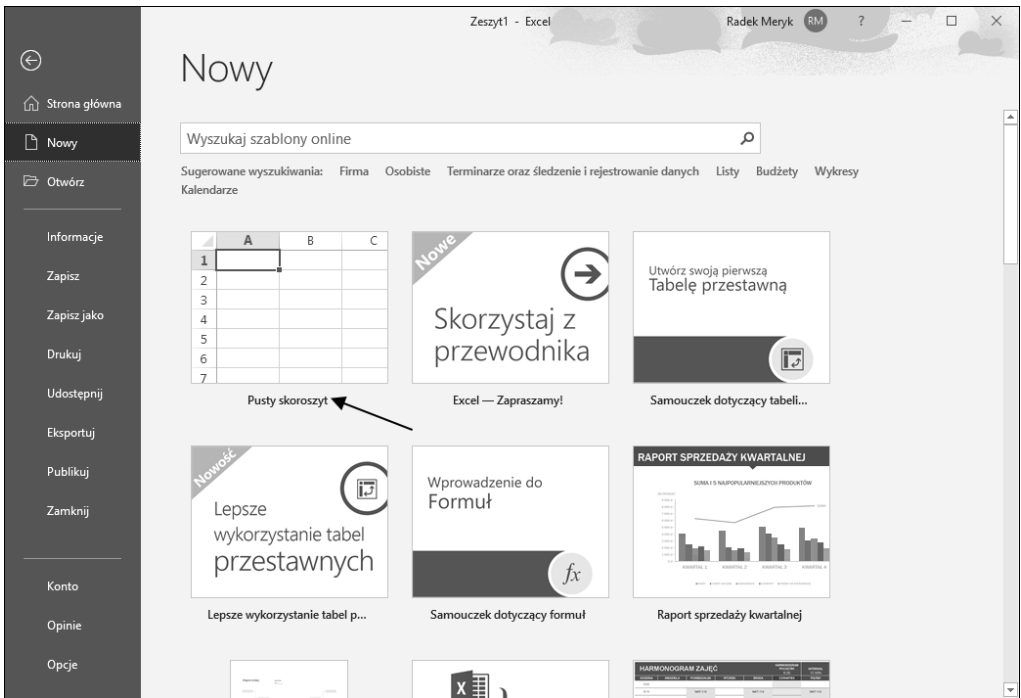
Rejestrowanie przykładowego makra w Excelu

W kolejnych punktach pokażemy, jak zarejestrować przykładowe makro Excela. To makro utworzy nowy skoroszyt, wprowadzi do niego sekwencję miesięcy, a następnie go zapisze. Z tym makrem będziemy ponownie pracować w rozdziale 3., więc go nie usuwaj.

Tworzenie skoroszytu makr osobistych, jeśli nie stworzono go wcześniej

Jeśli jeszcze nie masz skoroszytu makr osobistych w Excelu, będziesz musiał go utworzyć przed wykonaniem tej procedury (jeśli już masz skoroszyt makr osobistych, przejdź do następnego punktu). Wykonaj następujące czynności:

1. Uruchom Excela, a następnie kliknij pusty skoroszyt, tak aby była widoczna Wstążka (patrz rysunek 1.9).



RYСУNEK 1.9. Kliknięcie pustego skoroszytu

2. Kliknij najpierw zakładkę *Deweloper* na Wstążce, a następnie przycisk *Zarejestruj makro* (lub po prostu kliknij przycisk *Zarejestruj makro* na pasku stanu). Wyświetli się okno dialogowe *Rejestrowanie makra*.

Jeśli zakładka *Deweloper* nie jest widoczna, postępuj zgodnie z instrukcjami podanymi wcześniej w tym rozdziale, w punkcie zatytułowanym „Wyświetlanie zakładki *Deweloper* na Wstążce”. Jeśli opcje obsługi makr, takie jak przycisk *Zarejestruj makro*, są wyłączone — są szare i nie można ich kliknąć — postępuj zgodnie z instrukcjami zamieszczonymi w ramce pod tytułem „Uwaga na temat bezpieczeństwa” w tym rozdziale.

3. Zaakceptuj domyślną nazwę makra, ponieważ za chwilę i tak je usuniesz.
W tym momencie tylko tworzymy skoroszyt makr osobistych.
4. Na rozwijanej liście *Przechowuj makro* w wybierz *Skoroszyt makr osobistych*.
5. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Rejestrowanie makra* i rozpocząć nagrywanie makra.
6. Wpisz pojedynczy znak w aktywnej komórce, a następnie naciśnij klawisz *Enter*.
7. Kliknij przycisk *Zatrzymaj rejestrowanie* na Wstążce lub pasku stanu, aby zatrzymać rejestrowanie makra.
8. Kliknij przycisk *Odkryj* na zakładce *Widok*, aby wyświetlić okno dialogowe *Odkrywanie*.
9. Wybierz *PERSONAL* i kliknij przycisk *OK*.
10. Kliknij najpierw zakładkę *Deweloper* na Wstążce, a następnie przycisk *Makra*, aby wyświetlić okno dialogowe *Makra*.
11. Zaznacz makro, które zarejestrowałeś, a następnie kliknij przycisk *Usuń*, aby je usunąć.
12. Kliknij przycisk *Tak* w oknie komunikatu z potwierdzeniem.

Właśnie spowodowałeś utworzenie przez Excela skoroszytu makr osobistych. Odtąd możesz z niego korzystać do przechowywania globalnych makr.

Rejestrowanie makra

Teraz, aby stworzyć próbne makro, uruchom Excela i wykonaj następujące czynności:

1. Utwórz nowy skoroszyt, wybierając polecenie *Plik/Nowy/Pusty skoroszyt*.
2. Kliknij najpierw zakładkę *Deweloper* na Wstążce, a następnie przycisk *Zarejestruj makro* (lub po prostu kliknij przycisk nagrywania makra u dołu ekranu, na pasku stanu).

Spowoduje to wyświetlenie okna dialogowego *Rejestrowanie makra*, pokazanego na rysunku 1.3, z wprowadzonymi informacjami.

3. W polu tekstowym *Nazwa makra* wprowadź nazwę makra: *Dodaj_miesiace*.
4. Wciśnij, jeśli chcesz, klawisz skrót u w polu *Klawisz skrót*.
Pamiętaj, że zawsze można później zmienić klawisz skrót, więc nie musisz wprowadzać go teraz.

5. Na rozwijanej liście *Przechowuj makro* w zdecyduj, czy chcesz przechowywać makro w skoroszytcie makr osobistych, w nowym skoroszytcie, czy w aktywnym skoroszytcie.

Jak opisano nieco wcześniej w tym rozdziale, przechowywanie makr w skoroszytcie makr osobistych daje największą elastyczność, ponieważ jest to globalny kontener makr Excela. W tym przykładzie nie przechowuj makra w aktywnym skoroszytcie. Zamiast tego przechowaj je w skoroszytcie makr osobistych. Zapamiętaj, że będziemy używać tego makra w kolejnych przykładach, w dalszej części tej książki.

6. W polu tekstowym *Opis* wpisz opis makra.
7. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Rejestrowanie makra* i rozpocząć nagrywanie makra.

Kiedy Excel uruchamia się czasami, okazjonalnie, nie otwiera skoroszytu makr osobistych. Zatem jeśli zobaczysz komunikat o błędzie z informacją, że „Osobisty skoroszyt makr w folderze startowym musi pozostać otwarty do zapisu”, będziesz musiał otworzyć go ręcznie. Jeśli zobaczysz komunikat o podobnej treści, wykonaj następujące czynności:

- a. Zamknij okno dialogowe *Rejestrowanie makra*.
 - b. Wybierz *Plik/Opcje/Centrum zaufania*.
 - c. Kliknij przycisk *Ustawienia Centrum zaufania*.
 - d. W lewym panelu kliknij pozycję *Zaufane lokalizacje*.
 - e. Kliknij, aby wybrać opcję *Domyślna lokalizacja programu Excel: folder startowy użytkownika*, a następnie zwróć uwagę na dolną część okna, aby sprawdzić, gdzie jest ta lokalizacja. Powinna przypominać ścieżkę postaci `C:\Users\nazwa_użytkownika\AppData\Roaming\Microsoft\Excel\XLSTART`.
Użyj eksploratora plików, aby znaleźć tę ścieżkę, a następnie kliknij dwukrotnie plik *PERSONAL.XLSB*, aby go otworzyć.
 - f. Przejdź do kroku 8.
8. Kliknij komórkę *A1*, aby ją zaznaczyć.
Może już być zaznaczona; kliknij ją i tak, ponieważ chcemy zarejestrować to kliknięcie.
 9. Wpisz *Styczeń 2019* i naciśnij klawisz strzałki w prawo, aby zaznaczyć komórkę *B1*.
Excel automatycznie zmieni datę na swój domyślny format. Tak powinno być.
 10. Wpisz *Luty 2019*, a następnie naciśnij klawisz strzałki w lewo, aby ponownie zaznaczyć komórkę *A1*.
 11. Przeciągnij wskaźnik myszy od komórki *A1* do komórki *B1*, aby zaznaczyć obie.
 12. Przeciągnij uchwyt wypełniania od komórki *B1* do komórki *L1*, aby funkcja *Autouzupelnianie* Excela wprowadziła do komórek miesiące od marca do grudnia 2019.
Uchwyt wypełniania to mała czarna kropka w dolnym prawym rogu ramki wybierania. Będziesz wiedzieć, że nad nim jesteś, gdy kursor zmieni się z białego krzyżyka na czarny.
 13. Kliknij przycisk *Zatrzymaj rejestrowanie* na Wstążce.
Znajdziesz go po lewej stronie zakładki *Deweloper* — w sekcji *Kod*.

Teraz przetestujemy nasze nowe makro:

1. Utwórz nowy skoroszyt, wybierając polecenie *Plik/Nowy/Pusty skoroszyt*.
2. Na karcie *Deweloper* kliknij przycisk *Makra*, aby otworzyć okno dialogowe *Makro*.
3. Dwukrotnie kliknij pozycję *PERSONAL.XLSB!Dodaj_miesiące*.
Powinieneś zobaczyć, jak Excel wprowadza miesiące do nowego skoroszytu.

Usuń dwa skoroszyty, które utworzyłeś w tej sesji. Nie będą już potrzebne. Wystarczy kliknąć przycisk *X* w prawym górnym rogu okna, aby je zamknąć, a następnie wybrać opcję *Nie zapisuj*. Zapisz jednak skoroszyt *PERSONAL.XLST*. Wyświetli się komunikat z pytaniem, czy chcesz zapisać zmiany wprowadzone w skoroszytcie makr osobistych. Kliknij *Zapisz wszystko*.

Określanie sposobu uruchamiania istniejącego makra

Jeśli nie przypisałeś sposobu uruchamiania makra podczas jego nagrywania, możesz zrobić to teraz, postępując tak jak opisano w tym punkcie.

Przypisywanie makra do przycisku paska narzędzi Szybki dostęp w programie Word

Aby przypisać makro do paska narzędzi *Szybki dostęp*, wykonaj następujące czynności:

1. Kliknij prawym przyciskiem myszy w dowolnym miejscu na pasku narzędzi *Szybki dostęp* (jest to zbiór ikon w lewym górnym rogu ekranu, domyślnie powyżej *Wstążki*). Wyświetli się menu.
2. Z menu wybierz polecenie *Dostosuj pasek narzędzi Szybki dostęp*. Wyświetli się okno dialogowe *Opcje programu Word*.
3. Na liście rozwijanej *Wybierz polecenia z* wskaż pozycję *Makra*.
4. Kliknij nazwę makra, które chcesz przypisać do przycisku.
5. Kliknij przycisk *Dodaj*, aby skopiować nazwę makra do listy przycisków po prawej stronie.
6. Kliknij przycisk *Modyfikuj*, jeśli chcesz przypisać inną ikonę lub zmodyfikować nazwę przycisku.
7. Kliknij przycisk *OK*, aby zamknąć okno dialogowe.

Przypisywanie makra do klawisza skrót

W punkcie „Uruchamianie makra za pomocą kombinacji klawiszy skrót” znajdującym się wcześniej w tym rozdziale wyjaśniono, jak przypisać skrót do makra w programie Word. Programy PowerPoint i Access nie pozwalają na przypisywanie makr do kombinacji klawiszy. W Excelu stosowane jest nieco inne podejście niż w Wordzie — istnieje ograniczenie pozwalające stosować tylko kombinacje klawiszy *Ctrl* i *Shift*. Opisano to wcześniej w tym rozdziale, w punkcie „Przypisywanie sposobu uruchamiania makra w programie Excel”.

Usuwanie makra

Aby usunąć makro, które nie jest już potrzebne, wykonaj następujące czynności:

1. Kliknij *Deweloper/Makra*, aby wyświetlić okno dialogowe *Makra*.
2. Wybierz makro z listy *Nazwa makra*.
3. Kliknij przycisk *Usuń*.
4. W oknie z ostrzeżeniem, które się wyświetli, kliknij przycisk *Tak*.
5. Kliknij przycisk *Zamknij* lub *Anuluj*, aby zamknąć okno dialogowe *Makra*.

Możesz też skorzystać z następującej procedury:

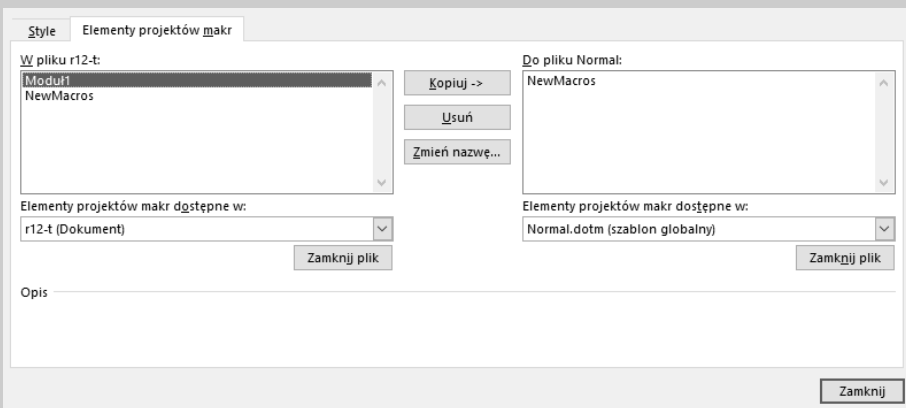
1. Znajdź makro w edytorze VBA.
2. Podświetl je, a następnie naciśnij klawisz *Del*, aby je usunąć.
3. Kliknij zakładkę *Deweloper*, a następnie kliknij ikonę Visual Basic, znajdującą się po lewej stronie *Wstążki*; otworzysz w ten sposób edytor.

ORGANIZOWANIE MAKRA W PROGRAMIE WORD ZA POMOCĄ OKNA DIALOGOWEGO ORGANIZATOR

Większość aplikacji obsługujących VBA w celu przeniesienia modułów kodu, formularzy użytkownika oraz innych elementów kodu z jednego pliku do innego, wymaga użycia programu Visual Basic Editor (opisanego szczegółowo w rozdziale 2.). Moduł kodu to wirtualny kontener, używany do przechowywania makr. Formularz użytkownika to spersonalizowane okno dialogowe, wyświetlane w celu pobrania od użytkownika danych wejściowych. Program Word udostępnia użyteczne narzędzie — okno *Organizator* — które można wykorzystać do kopiowania, przenoszenia, zmieniania nazwy i usuwania modułów kodu, formularzy użytkownika i innych elementów kodu bezpośrednio w interfejsie Worda, bez otwierania programu Visual Basic Editor.

Aby skorzystać z okna dialogowego *Organizator*, wykonaj następujące czynności:

1. Kliknij najpierw zakładkę *Deweloper*, a następnie ikonę *Makra*, drugą od lewej na Wstążce.
2. Kliknij przycisk *Organizator* w oknie dialogowym *Makra*.
3. Kliknij przycisk *Organizator*, aby wyświetlić okno dialogowe *Organizator*, a następnie kliknij zakładkę *Elementy projektów makr*, jeśli pokazana poniżej strona *Elementy projektów makra* nie wyświetli się automatycznie.



4. Spójrz na dwa dokumenty lub szablony wymienione w treści etykiet nad polami dwóch list. Zazwyczaj pole listy po lewej prezentuje aktywny dokument, a po prawej — szablon *Normal.dotm*.
5. Zmień to tak, aby jedna lista zawierała dokument lub szablon, który zawiera kod, który chcesz skopiować lub przenieść, a druga dokument lub szablon docelowy. Jeśli chcesz tylko usunąć fragment kodu lub zmienić nazwę elementu kodu, musisz wyświetlić w oknie dialogowym *Organizator* tylko ten dokument lub szablon, który zawiera elementy przeznaczone do modyfikacji.
6. Aby zmienić dokument lub szablon na liście, kliknij przycisk *Zamknij plik* pod polem listy na odpowiedniej stronie. Etykieta przycisku *Zamknij plik* zmieni się na *Otwórz plik*.
7. Kliknij przycisk *Otwórz plik*, aby wyświetlić okno dialogowe *Otwórz*. Przejdź do niego i wybierz żądany dokument lub szablon, a następnie kliknij przycisk *Otwórz*. Okno dialogowe *Otwórz* automatycznie domyślnie wyświetli folder *Szablony*.
8. Następnie możesz usunąć, zmienić nazwę, skopiować lub przenieść elementy projektu makra. Procedura zamieszczona poniżej prezentuje szczegółowe informacje na temat tego, jak to można zrobić:
 - ◆ Aby usunąć co najmniej jeden element projektu makra z szablonu, wybierz element (lub elementy) z dowolnego panelu okna dialogowego *Organizator* i kliknij przycisk *Usuń*. Kliknij przycisk *Tak* w oknie dialogowym z komunikatem potwierdzenia. Kopie elementów w innych szablonych pozostaną nienaruszone.

- ◆ Aby zmienić nazwę elementu projektu makra, zaznacz go w dowolnym panelu, a następnie kliknij przycisk *Zmień nazwę*. Otworzy się okno dialogowe *Zmianianie nazwy*. Wprowadź nową nazwę i kliknij przycisk *OK*. Kopie elementów w innych szablonach pozostaną nienaruszone.
 - ◆ Aby skopiować co najmniej jeden spośród elementów projektu makra z jednego szablonu do drugiego, otwórz szablon w oknie dialogowym *Organizator*. Wybierz element (lub elementy) do skopiowania w dowolnym panelu okna dialogowego (strzałki na przycisku *Kopij* zmieniają kierunek tak, aby wskazywał na długi panel). Następnie kliknij przycisk *Kopij*. Jeżeli szablon-odbiorca zawiera element projektu makra o tej samej nazwie jak ten, który kopiujesz, Word wyświetli okno z komunikatem ostrzegawczym, zawierającym informację, że nie można skopiować elementu. Jeśli jednak nadal chcesz go skopiować, zmień nazwę elementu, który kopiujesz albo tego o tej samej nazwie w szablonie docelowym, a następnie wykonaj operację kopiowania.
 - ◆ Aby przenieść element projektu makra z jednego szablonu do innego, skopiuj go w sposób opisany w poprzednim akapicie, a następnie usuń element projektu makra z szablonu źródłowego.
9. Po usunięciu, zmianie nazwy, skopiowaniu lub przeniesieniu elementu projektu makra, kliknij przycisk *Zamknij*, aby zamknąć okno dialogowe *Organizator*.
10. Jeśli program Word wyświetli monit, aby zapisać zmiany w tych dokumentach lub szablonach, na które mają wpływ wprowadzone zmiany, a które nie są otwarte w sesji programu Word, kliknij przycisk *Tak*.

Pytania kontrolne i zadania

Rejestrowanie makra. Najprostszym sposobem utworzenia makra jest jego zarejestrowanie. Wszystko, co wpiszesz lub klikniesz — wszystkie Twoje zachowania w programie — zostaną automatycznie przetłumaczone na język VBA i zapisane jako makro.

Zadanie. Włącz Rejestrator makr w programie Word i stwórz makro, które przesuwa kursor wstawiania o trzy wiersze w górę. Następnie wyłącz Rejestrator makr i przetestuj nowe makro.

Przypisywanie makra do przycisku lub klawisza skrót. Makro można wywołać za pomocą trzech wygodnych metod: kliknięcia elementu na Wstążce, kliknięcia przycisku na pasku narzędzi *Szybki dostęp* lub użycia skrótu klawiaturowego. Pamiętaj, aby przypisać do makra dowolną z tych metod — lub wszystkie.

Zadanie. Przypisz istniejące makro do nowego przycisku na pasku narzędzi *Szybki dostęp*.

Uruchamianie makra. Makra najłatwiej uruchomić za pośrednictwem Wstążki lub klikając przycisk na pasku narzędzi *Szybki dostęp*. Można też nacisnąć kombinację klawiszy skrótu, np. *Alt+N* lub *Ctrl+Alt+F*. Po rozpoczęciu rejestrowania makra otwiera się okno dialogowe *Rejestrowanie makra*, w którym są przyciski, pozwalające przypisać nowe makro do klawisza skrótu lub przycisku paska narzędzi. Jeśli jednak używasz programu Visual Basic Editor, możesz uruchomić makro przez naciśnięcie klawisza *F5*.

Zadanie. Uruchom makro z poziomu programu Visual Basic Editor.

Usuwanie makra. Warto zadbać o to, aby kolekcja makr była aktualna i możliwa do zarządzania. Jeśli nie potrzebujesz już makra, usuń je. Makra można usuwać bezpośrednio z programu Visual Basic Editor lub poprzez kliknięcie przycisku *Usuń* w oknie dialogowym *Makra* (aby je otworzyć, wybierz *Deweloper/Makra*).

Zadanie. Tymczasowo usuń makro, a następnie przywróć je za pomocą programu Visual Basic Editor.

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

VBA: zautomatyzuj nudną pracę i ciesz się tym, co lubisz!

VBA jest językiem programowania umożliwiającym automatyzację praktycznie wszystkich działań, które wykonuje się ręcznie w aplikacjach Office. Napisany w nim kod jest szybszy, dokładniejszy i bardziej niezawodny niż jakikolwiek człowiek. Można dodawać struktury decyzyjne i pętle, a tym samym znacznie poszerzyć zakres możliwych operacji. Ciekawą opcją jest też tworzenie w VBA interfejsów użytkownika — okien komunikatów, pól wprowadzania danych i formularzy. Fantastyczne możliwości zyskuje się w wyniku komunikowania się między sobą aplikacje pakietu Office: operacje w jednej aplikacji można wspomagać działaniami innej — za pomocą kodu VBA!

To praktyczny podręcznik do nauki programowania w języku VBA. Punktem wyjścia jest proces rejestracji makr oraz rozpoczęcie pracy w edytorze VBE. Dalej omówiono składnię języka VBA, potrzebne obiekty i metody oraz sposoby korzystania z pętli i funkcji, w tym funkcje wbudowane. Pokazano sposoby tworzenia interfejsów użytkownika zawierających takie elementy jak okna komunikatów, pola wprowadzania danych, niestandardowe okna dialogowe czy złożone formularze. Zaprezentowano także zasady budowania i debugowania modułowego kodu. W książce znalazło się wiele wskazówek oraz fragmentów kodu ułatwiających tworzenie przydatnych makr, dzięki którym praca z pakietem Office nabiera zupełnie innego kształtu!

W tej książce między innymi:

- podstawy pracy z VBA i korzystanie z Visual Basic Editor
- zmienne tablicowe, pętle i funkcje wbudowane
- pola wprowadzania danych i formularze
- tworzenie modułowego kodu i korzystanie z klas
- zabezpieczenia VBA i techniki zaawansowane

Richard Mansfield jest powszechnie znany jako ekspert w zakresie programowania.

Napisał ponad 40 książek poświęconych tej dziedzinie, wiele z nich zdobyło ogromną popularność i uznanie. W przeszłości był redaktorem magazynu „Computer!”.

Helion 	<i>Sprawdź nasze szkolenia!</i>	KOD KORZYŚCI <i>Sięgnij po więcej!</i> ▶	
 helion.pl	SZKOLENIA 	ISBN 978-83-283-6513-1	
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	AKADEMIA IT & BUSINESS HELIONSZKOLENIA.PL	 9 788328 365131	
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 149,00 zł	

 **SYBEX**
A Wiley Brand