

Automatyzacja w VBA dla Excela 2019

Receptury

Jak przyspieszać rutynowe zadania
i zwiększać efektywność pracy

Mike Van Niekerk

Helion 



Tytuł oryginału: VBA Automation for Excel 2019 Cookbook: Solutions to automate routine tasks and increase productivity with Excel and other MS Office applications

Tłumaczenie: Grzegorz Werner
ISBN: 978-83-283-9631-9

Copyright © Packt Publishing 2020. First published in the English language under the title 'VBA Automation for Excel 2019 Cookbook' – (9781789610031).

Polish edition copyright © 2022 by Helion S.A.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

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.

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

Autor oraz wydawca 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 wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/autvbe>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<https://ftp.helion.pl/przyklady/autvbe.zip>

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Printed in Poland.

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

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

Spis treści

| | |
|--|-----------|
| O autorze | 15 |
| O korektorze | 16 |
| Wstęp | 17 |
| Rozdział 1. Wprowadzenie do VBA | 23 |
| Wymagania techniczne | 23 |
| Analizowanie kodu VBA | 24 |
| Przygotowania | 24 |
| Realizacja | 25 |
| Działanie | 27 |
| Informacje dodatkowe | 27 |
| Rejestrowanie makra | 27 |
| Przygotowania | 28 |
| Realizacja | 28 |
| Działanie | 31 |
| Informacje dodatkowe | 31 |
| Testowanie makra | 31 |
| Przygotowania | 32 |
| Realizacja | 32 |
| Działanie | 33 |
| Informacje dodatkowe | 33 |
| Korzystanie z edytora VBA | 34 |
| Przygotowania | 34 |
| Realizacja | 34 |
| Działanie | 35 |
| Informacje dodatkowe | 35 |

| | |
|--|-----------|
| Edytowanie kodu przez zmianę odwołań do komórek | 35 |
| Przygotowania | 35 |
| Realizacja | 36 |
| Działanie | 37 |
| Informacje dodatkowe | 38 |
| Zapisywanie pliku z wbudowanym makrem | 38 |
| Przygotowania | 38 |
| Realizacja | 38 |
| Działanie | 39 |
| Informacje dodatkowe | 40 |
| Korzystanie z Centrum zaufania w kontekście bezpieczeństwa makr | 40 |
| Przygotowania | 40 |
| Realizacja | 40 |
| Działanie | 42 |
| Informacje dodatkowe | 42 |
| Tworzenie niestandardowej wstążki | 42 |
| Przygotowania | 42 |
| Realizacja | 43 |
| Rozdział 2. Korzystanie z edytora VBA | 50 |
| Wymagania techniczne | 50 |
| Identyfikowanie, włączanie i zamykanie składników edytora VBA | 51 |
| Przygotowania | 51 |
| Realizacja | 51 |
| Działanie | 52 |
| Tworzenie modułu do przechowywania procedury | 58 |
| Przygotowania | 58 |
| Realizacja | 58 |
| Działanie | 59 |
| Tworzenie prostej procedury w oknie kodu edytora | 59 |
| Przygotowania | 60 |
| Realizacja | 60 |
| Działanie | 61 |
| Tworzenie kodu VBA w oknie kodu na trzy różne sposoby | 61 |
| Przygotowania | 61 |
| Realizacja | 62 |
| Działanie | 65 |
| Informacje dodatkowe | 65 |
| Dostosowywanie edytora do własnych preferencji | 66 |
| Przygotowania | 66 |
| Realizacja | 66 |
| Działanie | 69 |
| Rozdział 3. Model obiektowy VBA | 71 |
| Wymagania techniczne | 71 |
| Istota obiektów w VBA | 72 |
| Przygotowania | 72 |
| Realizacja | 72 |
| Działanie | 72 |

| | |
|--|-----------|
| Obiekty jako kolekcje oraz modyfikowanie atrybutów całej kolekcji | 74 |
| Przygotowania | 74 |
| Realizacja | 74 |
| Działanie | 75 |
| Modyfikowanie właściwości obiektów | 76 |
| Przygotowania | 76 |
| Realizacja | 76 |
| Działanie | 77 |
| Informacje dodatkowe | 77 |
| Powiązanie metod z obiektami | 78 |
| Przygotowania | 79 |
| Realizacja | 79 |
| Działanie | 79 |
| Uaktywnianie obiektów za pomocą zdarzeń | 80 |
| Przygotowania | 80 |
| Realizacja | 80 |
| Działanie | 82 |
| Korzystanie z przeglądarki obiektów w edytorze VBA | 82 |
| Przygotowania | 83 |
| Realizacja | 83 |
| Działanie | 84 |
| Rozdział 4. Korzystanie z procedur | 85 |
| Wymagania techniczne | 86 |
| Tworzenie procedur i funkcji | 86 |
| Przygotowania | 86 |
| Realizacja | 86 |
| Działanie | 87 |
| Informacje dodatkowe | 88 |
| Bezpośrednie wykonywanie procedur | 89 |
| Przygotowania | 89 |
| Realizacja | 89 |
| Działanie | 90 |
| Wykonywanie procedur z poziomu okna dialogowego Makro | 90 |
| Przygotowania | 90 |
| Realizacja | 90 |
| Działanie | 91 |
| Wykonywanie procedury za pomocą przycisków | 91 |
| Przygotowania | 91 |
| Realizacja | 91 |
| Działanie | 93 |
| Informacje dodatkowe | 94 |
| Wykonywanie procedur za pomocą klawisza skrót | 94 |
| Przygotowania | 95 |
| Realizacja | 95 |
| Działanie | 96 |
| Wykonywanie funkcji za pomocą formuły w arkuszu | 96 |
| Przygotowania | 97 |
| Realizacja | 97 |
| Działanie | 97 |

| | |
|---|------------|
| Wykonywanie funkcji przez wywołanie jej wewnątrz procedury | 98 |
| Przygotowania | 98 |
| Realizacja | 98 |
| Działanie | 98 |
| Rozdział 5. Rejestrowanie zaawansowanych makr | 99 |
| Wymagania techniczne | 99 |
| Ustawienie odwołań bezwzględnych lub względnych w rejestratorze makr | 100 |
| Przygotowania | 100 |
| Realizacja | 100 |
| Działanie | 101 |
| Opcje rejestrowania | 102 |
| Przygotowania | 103 |
| Realizacja | 103 |
| Działanie | 103 |
| Rozdział 6. Elementy języka VBA | 105 |
| Wymagania techniczne | 106 |
| Dodawanie komentarzy | 106 |
| Przygotowania | 106 |
| Realizacja | 107 |
| Działanie | 107 |
| Deklarowanie zmiennych | 109 |
| Przygotowania | 110 |
| Realizacja | 110 |
| Działanie | 112 |
| Deklarowanie stałych | 116 |
| Przygotowania | 116 |
| Realizacja | 116 |
| Działanie | 117 |
| Informacje dodatkowe | 118 |
| Korzystanie ze zmiennych i stałych różnych typów | 119 |
| Przygotowania | 119 |
| Realizacja | 119 |
| Działanie | 120 |
| Definiowanie zasięgu zmiennych | 121 |
| Przygotowania | 122 |
| Realizacja | 122 |
| Działanie | 123 |
| Deklarowanie tablic | 124 |
| Przygotowania | 124 |
| Realizacja | 124 |
| Działanie | 127 |
| Rozdział 7. Korzystanie z zakresów | 131 |
| Wymagania techniczne | 131 |
| Odwołania do zakresów | 132 |
| Przygotowania | 132 |
| Realizacja | 133 |
| Działanie | 134 |

| | |
|--|------------|
| Korzystanie z właściwości obiektu Range | 134 |
| Przygotowania | 135 |
| Realizacja | 135 |
| Działanie | 135 |
| Stosowanie metod obiektu Range | 136 |
| Przygotowania | 137 |
| Realizacja | 137 |
| Działanie | 137 |
| Praktyczne zastosowanie zakresów | 138 |
| Przygotowania | 138 |
| Realizacja | 138 |
| Działanie | 140 |
| Rozdział 8. Stosowanie funkcji | 141 |
| Wymagania techniczne | 141 |
| Korzystanie z wbudowanych funkcji VBA | 141 |
| Przygotowania | 142 |
| Realizacja | 142 |
| Działanie | 144 |
| Stosowanie funkcji arkuszowych | 145 |
| Przygotowania | 145 |
| Realizacja | 146 |
| Działanie | 147 |
| Tworzenie własnych funkcji | 147 |
| Przygotowania | 147 |
| Realizacja | 147 |
| Działanie | 149 |
| Informacje dodatkowe | 149 |
| Rozdział 9. Sterowanie przebiegiem programu | 151 |
| Wymagania techniczne | 151 |
| Analiza przebiegu programu | 152 |
| Zmiana przebiegu za pomocą instrukcji GoTo | 152 |
| Przygotowania | 152 |
| Realizacja | 153 |
| Działanie | 153 |
| Podejmowanie decyzji z użyciem konstrukcji If i If-Then | 154 |
| Przygotowania | 154 |
| Realizacja | 154 |
| Działanie | 156 |
| Informacje dodatkowe | 156 |
| Konstrukcja Select Case | 157 |
| Przygotowania | 157 |
| Realizacja | 157 |
| Działanie | 158 |
| Stosowanie pętli | 161 |
| Przygotowania | 162 |
| Realizacja | 162 |
| Działanie | 166 |

| | |
|---|------------|
| Rozdział 10. Implementacja automatyzacji | 168 |
| Wymagania techniczne | 169 |
| Tworzenie procedur obsługi zdarzeń | 169 |
| Przygotowania | 169 |
| Realizacja | 169 |
| Działanie | 171 |
| Informacje dodatkowe | 172 |
| Poznanwanie zdarzeń | 173 |
| Obsługa zdarzeń skoroszytowych | 173 |
| Przygotowania | 174 |
| Realizacja | 174 |
| Działanie | 175 |
| Obsługa zdarzeń arkuszowych | 176 |
| Przygotowania | 176 |
| Realizacja | 176 |
| Działanie | 178 |
| Obsługa zdarzeń nieobiektywnych | 178 |
| Przygotowania | 179 |
| Realizacja | 179 |
| Działanie | 180 |
| Rozdział 11. Obsługa błędów | 181 |
| Wymagania techniczne | 181 |
| Obsługa błędów różnych rodzajów | 182 |
| Przygotowania | 182 |
| Realizacja | 182 |
| Działanie | 186 |
| Naprawianie błędu uruchomieniowego | 187 |
| Przygotowania | 187 |
| Realizacja | 187 |
| Działanie | 188 |
| Testowanie pod kątem wyszukiwania błędów | 189 |
| Przygotowania | 190 |
| Realizacja | 190 |
| Działanie | 193 |
| Stosowanie instrukcji On Error | 193 |
| Przygotowania | 194 |
| Realizacja | 194 |
| Działanie | 195 |
| Wznawianie działania po błędzie | 195 |
| Przygotowania | 195 |
| Realizacja | 195 |
| Działanie | 196 |

| | |
|---|------------|
| Rozdział 12. Debugowanie | 197 |
| Wymagania techniczne | 198 |
| Rodzaje błędów | 198 |
| Techniki debugowania | 199 |
| Przygotowania | 199 |
| Realizacja | 199 |
| Działanie | 203 |
| Stosowanie debugera w Excelu | 204 |
| Ustawianie punktów przerwania | 204 |
| Stosowanie okna Watches | 208 |
| Rozdział 13. Tworzenie i modyfikowanie okien dialogowych | 213 |
| Wymagania techniczne | 214 |
| Stosowanie funkcji MsgBox | 214 |
| Przygotowania | 214 |
| Realizacja | 214 |
| Działanie | 219 |
| Stosowanie funkcji InputBox | 220 |
| Przygotowania | 220 |
| Realizacja | 220 |
| Działanie | 223 |
| Stosowanie metody Application.InputBox | 223 |
| Przygotowania | 223 |
| Realizacja | 224 |
| Działanie | 227 |
| Rozdział 14. Tworzenie formularzy | 229 |
| Wymagania techniczne | 229 |
| Wstawianie formularza | 230 |
| Przygotowania | 230 |
| Realizacja | 230 |
| Działanie | 232 |
| Modyfikowanie właściwości | 232 |
| Przygotowania | 233 |
| Realizacja | 233 |
| Działanie | 235 |
| Dodawanie kontrolek | 236 |
| Przygotowania | 236 |
| Realizacja | 236 |
| Działanie | 240 |
| Informacje dodatkowe | 241 |
| Stosowanie ramek | 242 |
| Przygotowania | 242 |
| Realizacja | 242 |
| Informacje dodatkowe | 244 |

| | |
|--|------------|
| Rozdział 15. Kontrolki formularzy | 246 |
| Wymagania techniczne | 246 |
| Oprogramowanie kontrolek | 247 |
| Przygotowania | 247 |
| Realizacja | 247 |
| Działanie | 249 |
| Zamykanie formularza | 250 |
| Przygotowania | 250 |
| Realizacja | 250 |
| Działanie | 251 |
| Otwieranie formularza | 251 |
| Przygotowania | 251 |
| Realizacja | 252 |
| Działanie | 254 |
| Obsługa pola kombo | 254 |
| Przygotowania | 254 |
| Realizacja | 254 |
| Działanie | 256 |
| Obsługa pokrętki | 256 |
| Przygotowania | 256 |
| Realizacja | 257 |
| Działanie | 259 |
| Informacje dodatkowe | 259 |
| Ustalanie kolejności kontrolek | 259 |
| Przygotowania | 260 |
| Realizacja | 260 |
| Działanie | 261 |
| Rozdział 16. Tworzenie funkcji niestandardowych | 262 |
| Wymagania techniczne | 263 |
| Pisanie funkcji | 263 |
| Przygotowania | 263 |
| Realizacja | 263 |
| Działanie | 266 |
| Wywoływanie funkcji | 266 |
| Przygotowania | 266 |
| Realizacja | 266 |
| Działanie | 268 |
| Dodawanie argumentów | 268 |
| Przygotowania | 268 |
| Realizacja | 269 |
| Działanie | 270 |
| Informacje dodatkowe | 270 |
| Wyświetlanie opisu funkcji i jej argumentów | 271 |
| Przygotowania | 271 |
| Realizacja | 271 |

| | |
|---|------------|
| Rozdział 17. Tworzenie dokumentów Worda za pomocą VBA w Excelu | 275 |
| Wymagania techniczne | 276 |
| Tworzenie nowej instancji Worda | 276 |
| Przygotowania | 276 |
| Realizacja | 276 |
| Działanie | 278 |
| Wpisywanie i formatowanie tekstu | 279 |
| Przygotowania | 279 |
| Realizacja | 279 |
| Kopiowanie danych do Worda | 281 |
| Przygotowania | 281 |
| Realizacja | 281 |
| Działanie | 283 |
| Informacje dodatkowe | 284 |
| Stosowanie szablonów i zakładek | 284 |
| Przygotowania | 284 |
| Realizacja | 284 |
| Informacje dodatkowe | 286 |
| Rozdział 18. Generowanie plików PowerPointa z poziomu Excela | 287 |
| Wymagania techniczne | 287 |
| Tworzenie nowej instancji PowerPointa | 288 |
| Przygotowania | 288 |
| Realizacja | 288 |
| Generowanie prezentacji i slajdów | 290 |
| Przygotowania | 290 |
| Realizacja | 290 |
| Wstawianie treści w polach tekstowych | 291 |
| Przygotowania | 292 |
| Realizacja | 292 |
| Kopiowanie danych z Excela do PowerPointa | 293 |
| Przygotowania | 293 |
| Realizacja | 293 |
| Uzupełnienie | 295 |
| Skorowidz | 297 |

Wprowadzenie do VBA

Przepisy zawarte w tym rozdziale pomogą Ci ogólnie zapoznać się z językiem VBA. Dzięki nim poznasz środowisko robocze VBA i dowiesz się, jak tworzyć i zapisywać swoje własne procedury. Nauczysz się również konfigurować środowisko programistyczne i edytować kod VBA.

Przepisy prezentowane w tym rozdziale obejmują następujące zagadnienia:

- analizowanie kodu VBA;
- rejestrowanie makra;
- testowanie makra;
- korzystanie z edytora VBA;
- edytowanie kodu przez zmianę odwołań do komórek;
- zapisywanie pliku z wbudowanym makrem;
- korzystanie z Centrum zaufania w kontekście bezpieczeństwa makr;
- tworzenie niestandardowej wstążki.

Po przestudiowaniu tego rozdziału będziesz wiedzieć, do czego służą makra, w jaki sposób należy je rejestrować, jak edytować kod VBA oraz jak zapisać makro w skrószycie programu Excel.

Wymagania techniczne

Z niniejszej książki należy korzystać wraz z programem MS Office 2019 lub MS Office 365 zainstalowanym w systemie Windows 8, 8.1 lub 10.

W aplikacjach MS Word, MS Excel i MS PowerPoint można na wstążce włączyć wyświetlanie karty *Developer*, choć nie jest to bezwzględnie konieczne. Każdy z tych programów zawiera kartę *Widok* z grupą *Makra*, na której znajdują się ikony poleceń wyświetlania i rejestrowania makr.

Pliki demonstracyjne są dostępne pod adresem <https://ftp.helion.pl/przyklady/autvbe.zip>.

Na stronie <https://bit.ly/3jQRvVk> znajdziesz filmy z serii „Kod w akcji” (ang. *Code in Action*).

Analizowanie kodu VBA

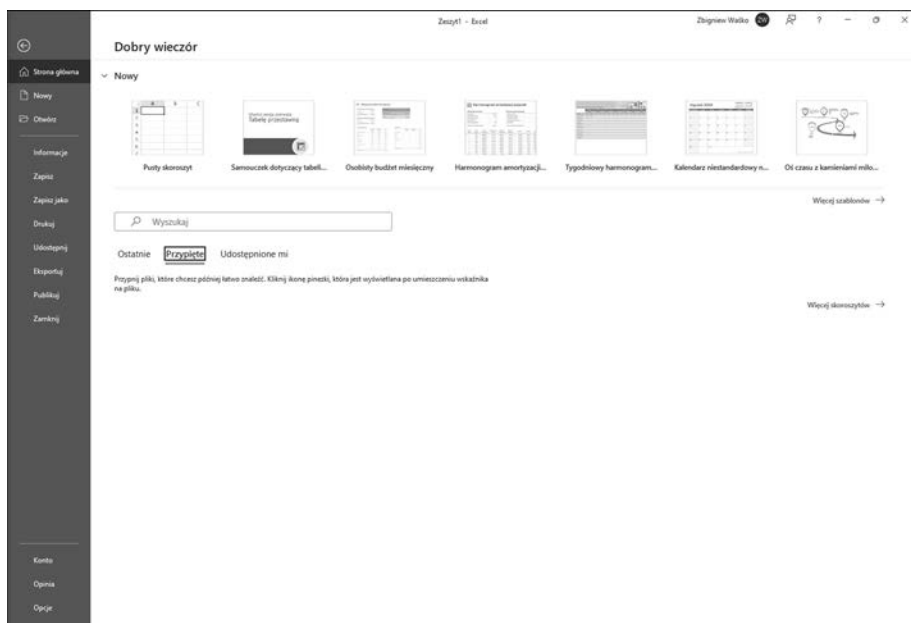
W tym przepisie przeanalizujemy krótką procedurę VBA. Podobnie jak wszystkie języki programowania także VBA ma właściwą dla siebie składnię. Najlepszym sposobem na poznanie jej zasad jest przyjrzenie się, jak wygląda konkretny kod.

Zagadnienia poruszone w tym rozdziale będą stanowić podstawę do zrozumienia wszystkich przepisów w książce, dlatego warto zwrócić na nie szczególną uwagę.

Przygotowania

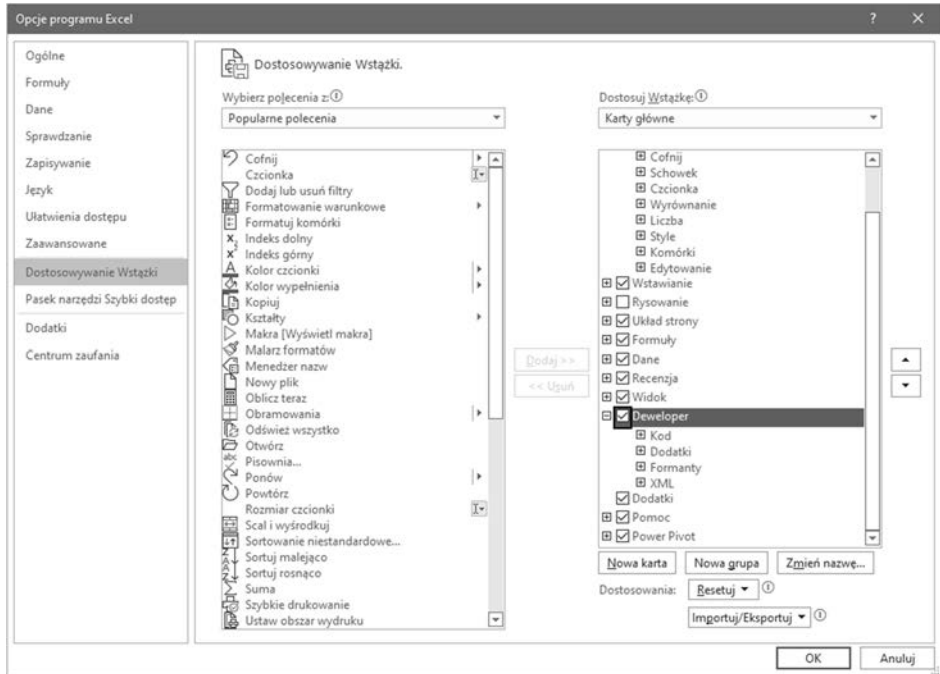
Aby móc analizować i edytować kod VBA w Excelu lub jakiegokolwiek innej aplikacji pakietu Office, najpierw należy włączyć na wstążce kartę *Developer*. W tym celu:

1. Otwórz Excela i na ekranie powitalnym wybierz opcję *Pusty skoroszyt*.
2. Otwórz plik demonstracyjny *01_VBA_Code.xlsm*. Na pasku komunikatów **WIDOK CHRONIONY** kliknij przycisk *Włącz edytowanie*.
3. Jeśli nie widzisz karty *Developer*, kliknij zakładkę *Plik*, aby włączyć widok *Backstage*. Okno Excela przyjmie formę pokazaną na rysunku 1.1.



Rysunek 1.1. Widok Backstage

4. Z listy kategorii po lewej stronie wybierz ostatnią pozycję, o nazwie *Opcje*.
Otworzy się okno dialogowe *Opcje programu Excel* pokazane na rysunku 1.2.



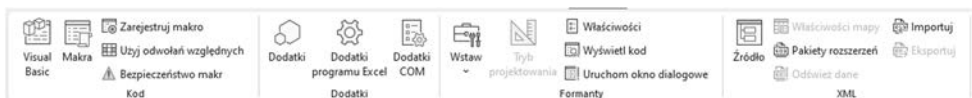
Rysunek 1.2. Okno dialogowe Opcje programu Excel

5. Z kategorii po lewej stronie wybierz *Dostosowywanie Wstążki*.
6. Na liście *Karty główne* po prawej stronie znajdź pozycję *Developer*. Zaznacz pole wyboru.
7. Kliknij przycisk *OK*, aby zatwierdzić zmianę. Okno zostanie zamknięte, a Excel doda do wstążki kartę *Developer*.

Realizacja

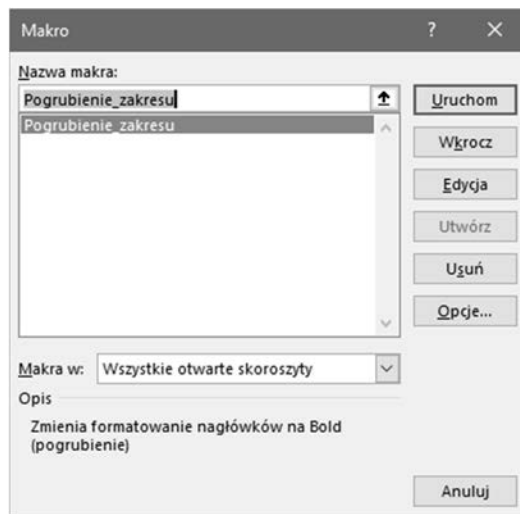
Gdy karta *Developer* jest już aktywna, możemy przejść do realizacji przepisu:

1. Przy otwartym pliku *01_VBA_Code.xlsm* kliknij zakładkę *Developer*, której zawartość jest pokazana na rysunku 1.3.



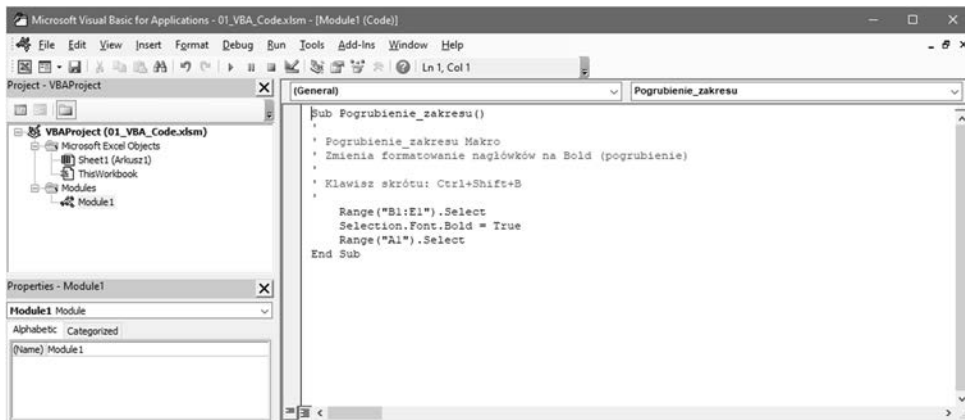
Rysunek 1.3. Karta Developer

2. W grupie *Kod* (po lewej stronie wstążki) kliknij ikonę *Makra*. Pojawi się okno o nazwie *Makro* pokazane na rysunku 1.4.



Rysunek 1.4. Okno dialogowe Makro

3. Analizowany kod VBA jest zapisany w jednym dostępnym makrze: *Pogrubienie_zakresu*. Kliknij przycisk *Edycja*. Otworzy się okno *Microsoft Visual Basic for Applications* pokazane na rysunku 1.5. W razie potrzeby zmaksymalizuj jego rozmiar.



Rysunek 1.5. Okno edytora Microsoft Visual Basic for Applications

4. W oknie kodu (duży obszar po prawej stronie) wyświetlana jest krótką procedura VBA.

Działanie

Procedura w edytorze VBA wygląda tak jak na rysunku 1.6.

```
Sub Pogrubienie_zakresu()
|
|
| ' Pogrubienie_zakresu Makro
| ' Zmienia formatowanie nagłówków na Bold (pogrubienie)
|
|
| ' Klawisz skrótu: Ctrl+Shift+B
|
|
|     Range("B1:E1").Select
|     Selection.Font.Bold = True
|     Range("A1").Select
| End Sub
```

Rysunek 1.6. Procedura w oknie edytora VBA

Spróbujmy zrozumieć ten podprogram. Wszystkie procedury w VBA zaczynają się od słowa kluczowego `Sub`, po którym następuje nazwa podprogramu (makra), a kończą się dwoma nawiasami.

Pojedynczy apostrof pozwala na wprowadzenie uwagi. Komentarze, które są w kolorze zielonym, można wstawiać w dowolnym miejscu procedury, pod warunkiem że znajdują się w osobnym wierszu lub za tekstem kodu (tzn. nie mogą znajdować się przed tekstem kodu, ponieważ w takiej sytuacji także kod stałby się komentarzem). Wiersze z wcięciem i bez apostrofów oznaczają instrukcje VBA. W tym przypadku pierwsza instrukcja nakazuje zaznaczenie komórek z zakresu od B1 do E1.

Następnie ustawiany jest styl czcionki jako **Bold** (pogrubienie). Ostatnia instrukcja zaznacza komórkę A1. Procedura kończy się słowami kluczowymi `End Sub`.

Informacje dodatkowe

Niezależnie od tego, czy nagrywasz makro, czy ręcznie wprowadzasz kod, każdy program VBA będzie miał taką podstawową strukturę składniową jak ta opisana powyżej.

W następnych rozdziałach będziemy mieć do czynienia z o wiele dłuższymi programami. Zanim jednak do tego dojdziemy, spróbujmy zarejestrować makro. Dzięki nowo nabytej wiedzy będziesz w stanie przeanalizować również tak powstały kod VBA.

Rejestrowanie makra

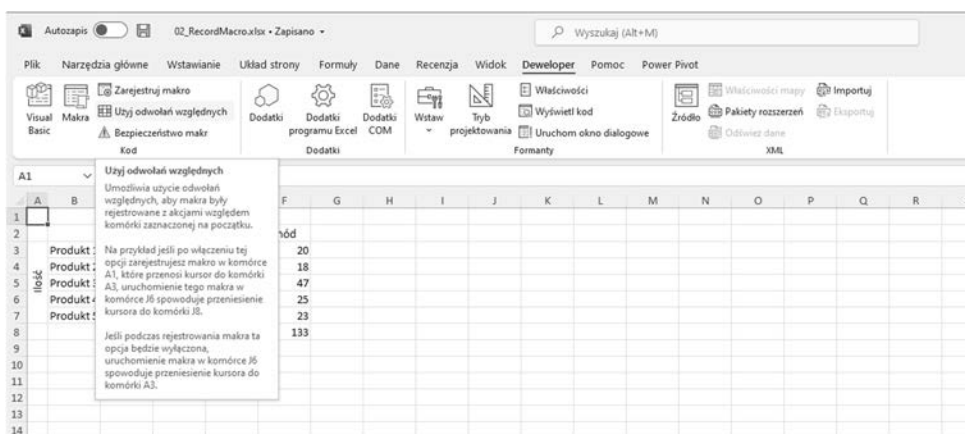
Makra są niezastąpione, gdy musimy wykonywać powtarzające się zadania. W tym przepisie utworzymy takie właśnie makro. Zamiast ręcznie wpisywać wiele wierszy kodu VBA w celu utworzenia procedury lub krótkiego programu, możemy po prostu nagrać ciąg czynności w Excelu, a następnie zapisać je w tym samym pliku co dane. Rejestrator makr automatycznie utworzy kod VBA — przekonasz się o tym w następnych przepisach.

Nagrane czynności można następnie odtwarzać, aby w ułamku sekundy uzyskać efekt wymagający wykonania szeregu mniej lub bardziej złożonych operacji.

Przygotowania

Podobnie jak w funkcjach również w makrach wykorzystuje się odwołania bezwzględne i względne. W pierwszym makrze użyjemy odwołań bezwzględnych, ponieważ mamy do czynienia z dwoma arkuszami o dokładnie takim samym układzie.

Otwórz plik *02_RecordMacro.xlsx* i sprawdź, czy aktywny jest arkusz pierwszy. Kliknij kartę *Developer*. Upewnij się, że opcja *Użyj odwołań względnych* w grupie *Kod* nie jest aktywna (zobacz rysunek 1.7).

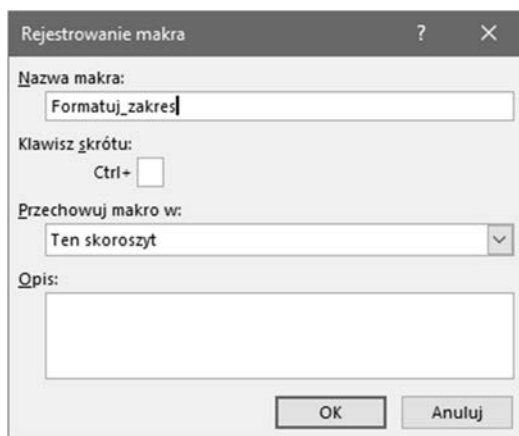


Rysunek 1.7. Opcja Odwołania względne

Realizacja

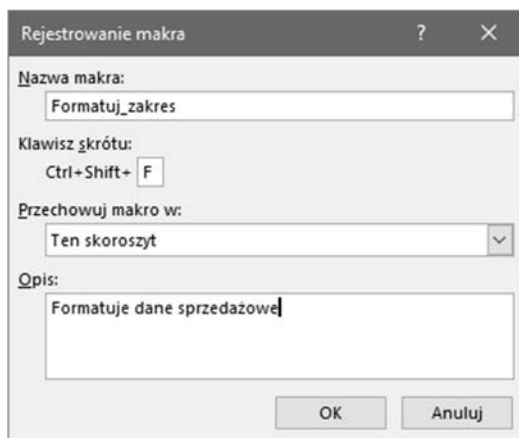
Wykonamy teraz procedurę rejestrowania makra:

1. Sprawdź, czy w pliku *02_RecordMacro.xlsx* aktywny jest arkusz pierwszy i czy zaznaczona jest komórka A1. W grupie *Kod* na karcie *Developer* kliknij opcję *Zarejestruj makro*. Otworzy się okno dialogowe *Rejestrowanie makra*.
2. W pierwszym polu tekstowym, *Nazwa makra*, wpisz *Formatuj_zakres*. Będzie to nazwa rejestrowanego makra.
3. Naciśnij klawisz *Tab* lub kliknij pole tekstowe znajdujące się pod nagłówkiem *Klawisz skrót* i na prawo od tekstu *Ctrl +* (zobacz rysunek 1.8).
4. Trzymając naciśnięty klawisz *Shift*, naciśnij klawisz *F*.
5. Skrót dla tego makra zostanie ustawiony jako kombinacja *Ctrl + Shift + F*.



Rysunek 1.8. Pole tekstowe skrótu klawiszowego, Ctrl +

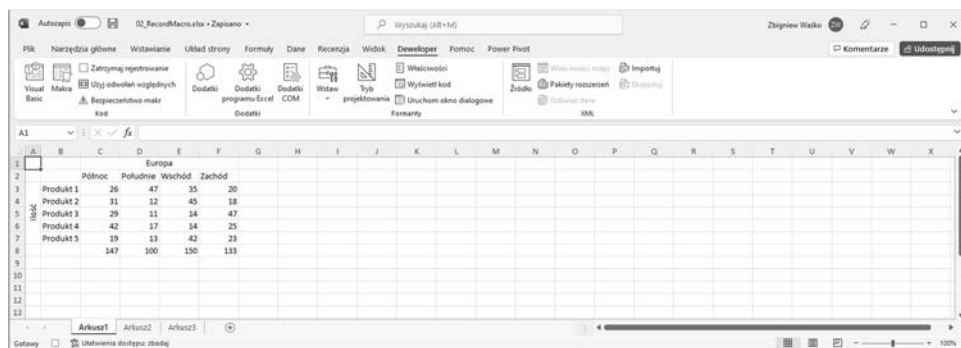
6. Na liście rozwijanej *Przechowuj makro w* pozostaw wybraną pozycję *Ten skoroszyt*. Pozostałe pozycje tej listy zostaną omówione w następnych przepisach.
7. W polu *Opis* wpisz krótkie objaśnienie, do czego makro będzie służyć. Kliknij *OK*, aby rozpocząć rejestrowanie (zobacz rysunek 1.9).



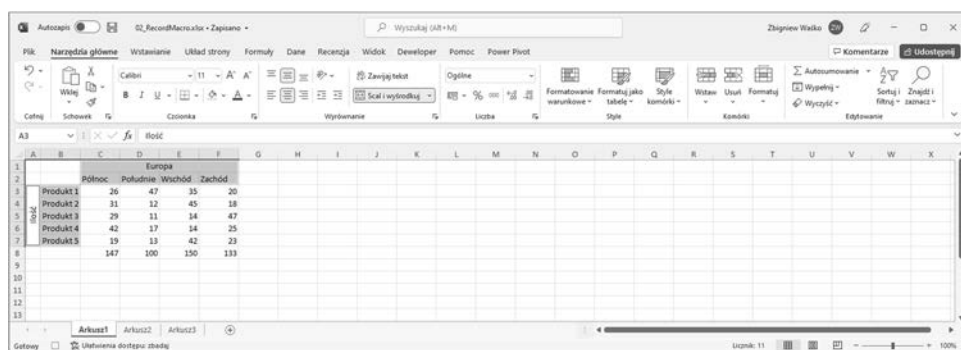
Rysunek 1.9. Okno dialogowe Rejestrowanie makra z wypełnionymi wszystkimi polami

8. Zwróć uwagę na grupę *Kod* na karcie *Developer*. Ikona *Zarejestruj makro* zmieniła się na ikonę *Zatrzymaj rejestrowanie*. Oznacza to, że jesteś w trybie rejestrowania, czyli wszystko, co zrobisz za pomocą klawiatury i myszy, zostanie nagrane (zobacz rysunek 1.10).
9. W trybie rejestrowania otwórz kartę *Narzędzia główne*. Następnie zaznacz zakres komórek od C1 do F2, po czym przytrzymaj klawisz *Ctrl* i zaznacz zakres od A3 do B7 (zobacz rysunek 1.11).

Zmień teraz format wybranych komórek na *Pogrubienie*.

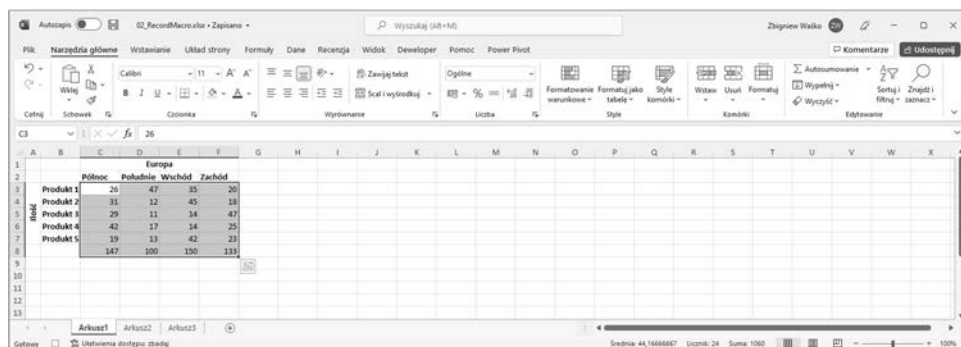


Rysunek 1.10. W trybie nagrywania wyświetlana jest ikona Zatrzymaj rejestrowanie



Rysunek 1.11. Zaznaczone zakresy

10. Następnie zaznacz komórki z zakresu od C3 do F8 (zobacz rysunek 1.12).



Rysunek 1.12. Zaznaczony zakres

Wyśrodkuj zawartość komórek i ustaw format walutowy (zł).

11. Na koniec zaznacz komórkę A1, po czym otwórz kartę *Developer* i w grupie *Kod* kliknij ikonę *Zatrzymaj rejestrowanie*.

Właśnie udało Ci się zapisać makro.

Działanie

Prawda, że nie było to takie trudne? Przyjrzyjmy się teraz, co tak właściwie się stało.

Ustawiliśmy odniesienia dla makra jako bezwzględne, ponieważ to samo makro ma się odnosić także do pozostałych arkuszy (*Arkusz2* i *Arkusz3*). Jeśli uruchomisz to makro w dowolnym z tych arkuszy, uzyskasz za każdym razem taki sam efekt.

Nazwy makr muszą zaczynać się od liter i nie mogą zawierać spacji. Dodatkowo nie mogą one być słowami kluczowymi VBA, takimi jak *Format*, *Copy* czy *Select*. Zabronione jest również stosowanie znaków specjalnych.

Skrót klawiszowy ułatwia uruchamianie makra, ale zgodnie z nazwą jest tylko skrótem. W następnym przepisie uruchomimy makro z poziomu okna *Makro*.

Zapisanie makra z opcją *W tym skoroszytcie* oznacza, że będzie ono działać tylko w obrębie konkretnego arkusza. Aby było dostępne w innych skoroszytach, należy zastosować opcję *Skoroszyt makr osobistych*. Więcej informacji na ten temat znajdziesz w kolejnych przepisach.

Wielu użytkowników nie wypełnia pola *Opis*. Nie wpływa to na samo makro, ale warto wyjaśnić innym użytkownikom, jakie zadania ma ono do wykonania.

W trybie rejestrowania zapisywane są wszystkie działania, włącznie z błędnymi. Dobrym zwyczajem jest więc wyszczególnienie kroków, które chcemy zarejestrować.

Po kliknięciu przycisku *Zatrzymaj rejestrowanie* rejestrator zostanie wyłączony, a Ty możesz na chwilę się odprężyć. Wszystkie Twoje działania zostały zapisane i przekształcone na kod VBA. Gdy otworzysz edytor *Microsoft Visual Basic for Applications*, zobaczysz kod zapisany w trybie rejestrowania.

Informacje dodatkowe

Gdy już przekonasz się, że rejestrowanie całego ciągu czynności w Excelu jest naprawdę łatwe, zaczniesz szukać możliwości automatyzowania wszystkich powtarzających się zadań.

Możesz na przykład zapisać makro polegające na zaznaczeniu całego arkusza i usunięciu formatowania. Będzie to znacznie szybsze niż ręczne wykonywanie tych operacji, zwłaszcza jeśli plik zawiera więcej niż jeden arkusz.

Testowanie makra

Samo zarejestrowanie makra to już coś, ale warto się przekonać, czy ono działa. Jest na to tylko jeden sposób — trzeba je przetestować. Oczywiście możesz także sprawdzić kod VBA

w edytorze, ale do momentu, gdy nie staniiesz się doświadczonym programistą, znalezienie błędów w kodzie będzie praktycznie niemożliwe. Najlepiej więc wykonać praktyczny test.

Przygotowania

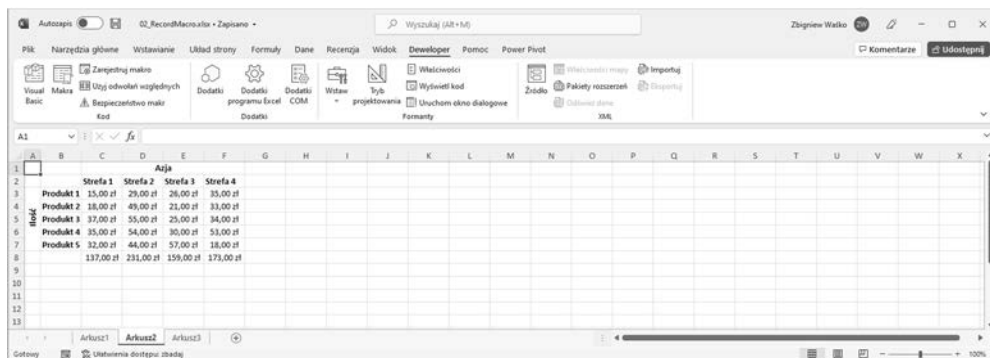
Zanim przejdziemy dalej, upewnij się, że makro zostało zarejestrowane zgodnie z instrukcjami z poprzedniego przepisu. Będziemy bowiem teraz bazować na wygenerowanym wtedy kodzie.

Makro zostało zarejestrowane na arkuszu pierwszym, dlatego teraz spróbujemy je przetestować na arkuszu drugim. Sprawdzanie na pierwszym nie ma sensu, ponieważ komórki są tam już odpowiednio sformatowane. Drugi arkusz ma nadal pierwotną postać, dlatego wszystkie zmiany będą natychmiast widoczne.

Realizacja

Wykonaj następujące czynności:

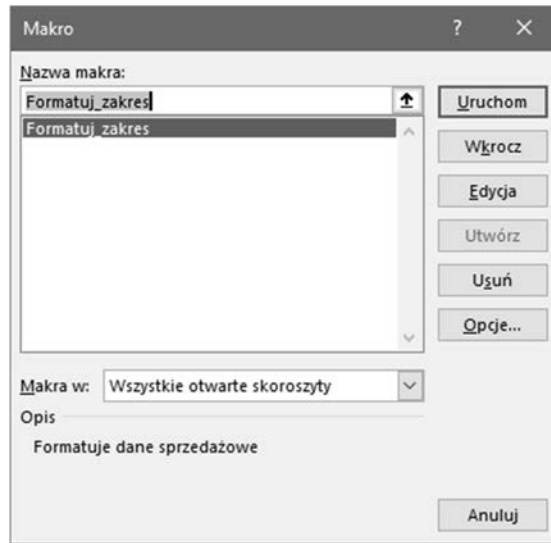
1. Upewnij się, że wciąż otwarty jest plik *02_RecordMacro.xlsx*, ale tym razem wybierz arkusz drugi. Aby sprawdzić działanie ustawienia *Odwolanie względne*, wybierz dowolną komórkę inną niż A1.
2. Naciśnij teraz kombinację klawiszy *Ctrl + Shift + F* i zobacz, co się stanie (zobacz rysunek 1.13).



Rysunek 1.13. Arkusz sformatowany po uruchomieniu makra

Zawartość komórek zostanie sformatowana dokładnie tak jak na arkuszu pierwszym. Prawda, że to znacznie szybszy sposób niż ręczne formatowanie każdego obszaru osobno?

3. Otwórz arkusz trzeci. Zobaczysz, że jest jeszcze jeden sposób uruchamiania makra. Wybierz dowolną komórkę arkusza.
4. Otwórz kartę *Developer* i w grupie *Kod* kliknij ikonę *Makra*, aby otworzyć okno dialogowe *Makro* pokazane na rysunku 1.14.



Rysunek 1.14. Okno dialogowe Makro

5. Jak widać jedynym dostępnym makrem jest to, które już zarejestrowaliśmy. Po prawej stronie znajduje się kilka przycisków. Kliknij ten u samej góry — o nazwie *Uruchom*.
6. Zgodnie z oczekiwaniami arkusz trzeci zostanie sformatowany w taki sam sposób jak pierwszy i drugi.

Działanie

Gdy nagrywaliśmy makro, w oknie dialogowym *Rejestrowanie makra* dostępna była opcja ustawienia skrótu klawiszowego. Najwyraźniej nasze instrukcje zadziałały.

Nieco dłuższym, ale nadal skutecznym sposobem jest uruchomienie makra z okna dialogowego *Makro*. Ponieważ istnieje tylko jedno makro, możemy tylko to jedno wykonać. Gdyby jednak było ich więcej, moglibyśmy wybrać dowolne z nich i nacisnąć przycisk *Uruchom*.

Informacje dodatkowe

Dobłą praktyką jest przetestowanie makra zamiast zakładania, że wszystko jest w porządku i musi zadziałać. Bardzo rzadko się zdarza, że zarejestrowane makro nie działa, ale kiedy przejdziemy do bardziej zaawansowanych procedur — zwłaszcza gdy zaczniemy je pisać ręcznie — nigdy nie powinniśmy zakładać, że nasz program będzie bezbłędny.

Skoro już wiemy, że wszystko działa tak, jak powinno, warto zobaczyć, jak wygląda rzeczywisty kod VBA.

Korzystanie z edytora VBA

Z edytora VBA korzystaliśmy już wcześniej, ale dotyczyło to innego pliku i innej procedury. Przepis, którym teraz się zajmiemy, pokaże, jak uaktywnić edytor VBA z poziomu otwartego pliku. Postaramy się również zidentyfikować poszczególne elementy edytora.

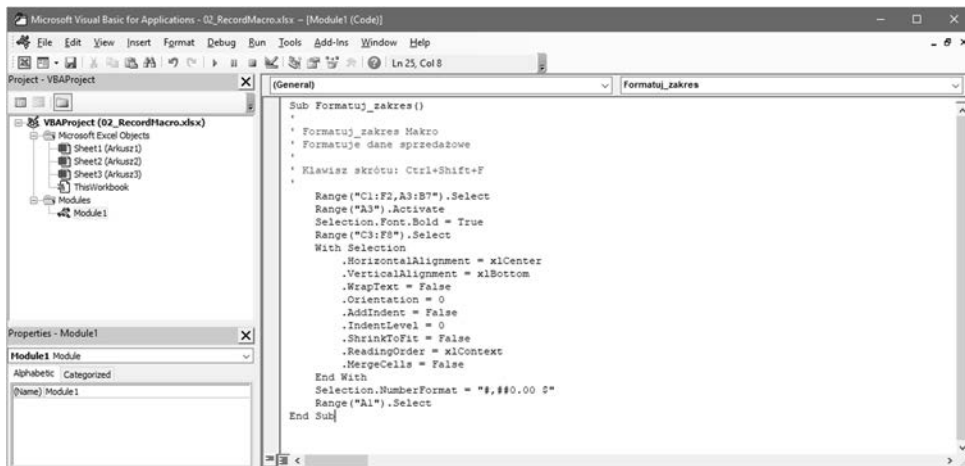
Przygotowania

Przy nadal otwartym pliku *02_RecordMacro.xlsx* spróbujemy włączyć edytor VBA. To, czy otwarty jest arkusz pierwszy, drugi czy trzeci, tak naprawdę nie ma znaczenia, ponieważ makro stanowi element całego skoroszytu.

Realizacja

Wykonaj teraz poszczególne etapy przepisu:

1. Klasyčnym sposobem uruchamiania edytora VBA jest wybranie opcji *Makra* w grupie *Kod* na karcie *Developer*, a następnie kliknięcie przycisku *Edycja* w oknie *Makro*.
2. Skrót klawiszowy znacznie ułatwia życie w tym względzie. Przytrzymaj klawisz *Alt* i naciśnij *F11*. Tym sposobem pominiemy okno dialogowe i przejdziesz bezpośrednio do edytora VBA (zobacz rysunek 1.15).



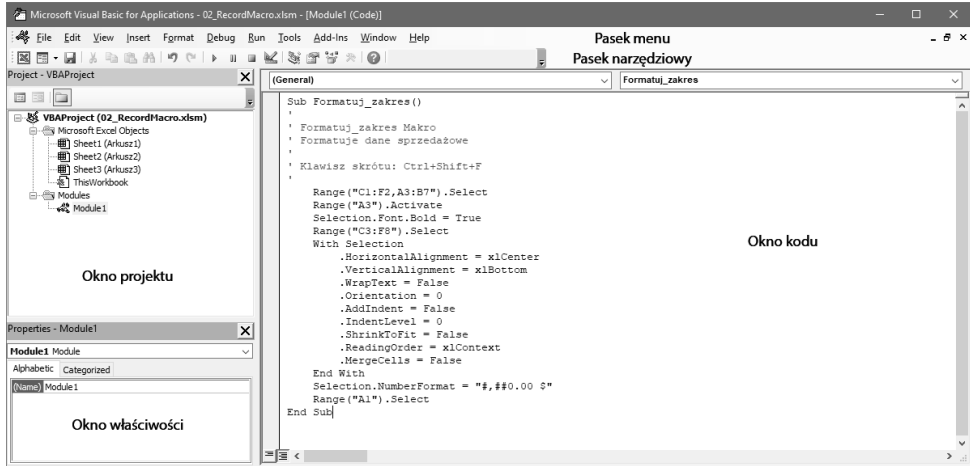
Rysunek 1.15. Kod procedury wyświetlany w edytorze VBA

To, co zarejestrowaliśmy, jest wyświetlane w oknie kodu.

Działanie

Edytor VBA można otworzyć na dwa sposoby: za pomocą skrótu klawiszowego *Alt + F11* lub przez kliknięcie przycisku *Edycja* w oknie dialogowym *Makro*.

Podstawowe komponenty edytora są pokazane na rysunku 1.16.



Rysunek 1.16. Komponenty edytora VBA

Informacje dodatkowe

Każdy komponent edytora VBA pełni określoną funkcję. Do tej pory korzystaliśmy tylko z okna kodu. W kolejnych przepisach będziemy używać także innych elementów.

Edytowanie kodu przez zmianę odwołań do komórek

W tym przepisie pokazałem, w jaki sposób edytować kod w edytorze VBA. Bardziej efektywne jest wprowadzanie zmian w kodzie niż usuwanie całego makra i nagrywanie go od nowa. Zaczniemy od usunięcia kilku zbędnych wierszy, a następnie nauczysz się, jak edytować kod w taki sposób, aby wywołać widoczną zmianę w arkuszu.

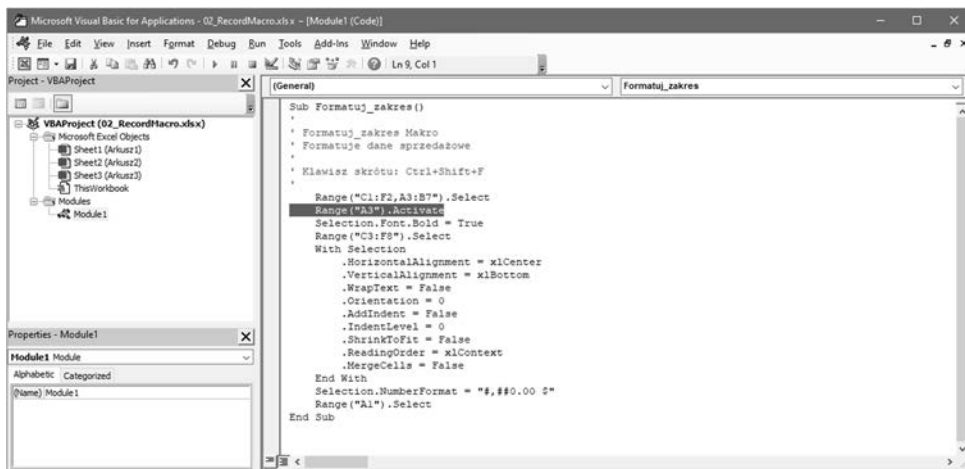
Przygotowania

Przy nadal otwartym pliku *02_RecordMacro.xlsx* naciśnij klawisze *Alt + F11*, aby otworzyć edytor VBA.

Realizacja

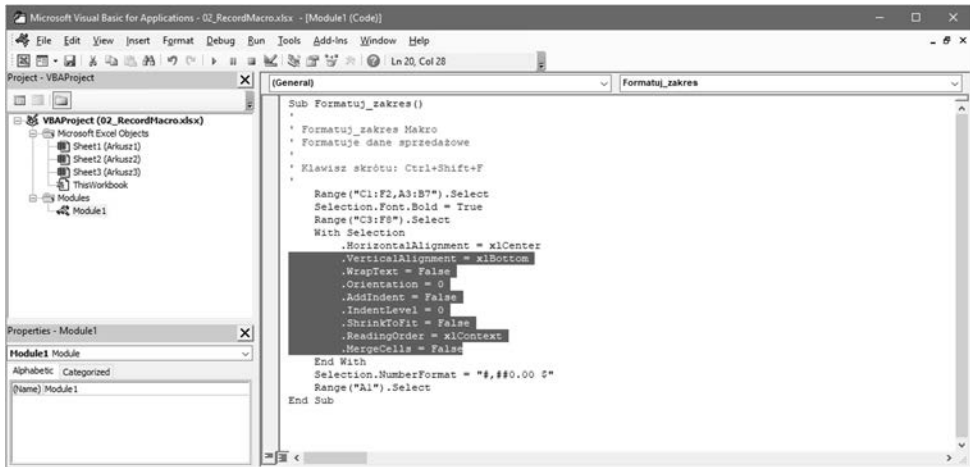
Ten przepis obejmuje następujące czynności:

1. W edytorze VBA dwukrotnie kliknij pozycję *Module1* (moduł1), aby otworzyć okno kodu.
2. Zaznacz drugi wiersz wykonywalnego kodu. Usuń zarówno wiersz kodu, jak i pusty wiersz (zobacz rysunek 1.17).

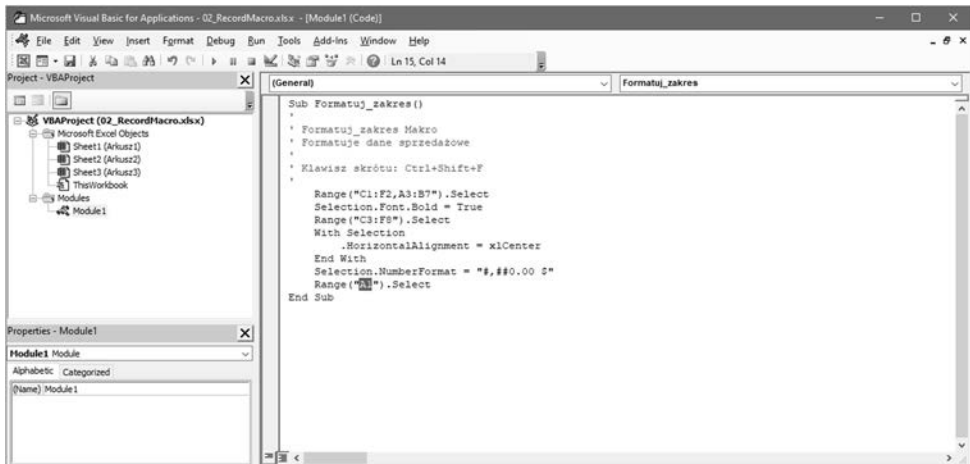


Rysunek 1.17. Kod zaznaczony do usunięcia

3. Powróć do Excela, usuń formatowanie danych i uruchom makro. Wynik będzie taki sam jak po pierwszym teście.
4. Następnie zaznacz wiersze znajdujące się poniżej tekstu `HorizontalAlignment = xlCenter` (zobacz rysunek 1.18) i naciśnij klawisz *Del*.
5. Ponownie wróć do Excela, usuń formatowanie danych i uruchom makro. Wynik znowu będzie taki sam jak po pierwszym teście.
6. Na koniec zmień odniesienie do komórki w ostatnim wierszu z A1 na G2 (zobacz rysunek 1.19).
7. Po raz ostatni wróć do Excela, usuń formatowanie danych w arkuszu i uruchom makro. Komórki zostaną sformatowane tak jak poprzednio, ale ostatecznie zostanie zaznaczona komórka G2, a nie A1.



Rysunek 1.18. Kod zaznaczony do usunięcia



Rysunek 1.19. Zmiana odniesienia do komórki

Działanie

Spróbujmy zastanowić się, co właśnie zrobiliśmy.

Wiersz `Range("A3").Activate` można usunąć, ponieważ zakresy C1:F2 i A3:B7 są już zaznaczone. Są to komórki przeznaczone do pogrubienia.

W przypadku zakresu C3:F8 zależy nam jedynie na wyśrodkowaniu zawartości. Pozostałe linie są dodawane automatycznie, ale są niepotrzebne, dlatego można je usunąć.

I wreszcie zamiast zakończenia procedury na komórce A1 wybraliśmy G2.

Informacje dodatkowe

Makro zostało już zarejestrowane, przetestowane i poddane edycji, ale nie zapisaliśmy jeszcze pliku. Być może brzmi to banalnie, ale w rzeczywistości nie jest to taka oczywista operacja. Z kolejnego przepisu dowiesz się, dlaczego.

Zapisywanie pliku z wbudowanym makrem

W tym przepisie nauczysz się, jak zapisać plik z wbudowanym makrem. Pewnie myślisz, że zapisanie pliku nie może być takie trudne. I tak naprawdę nie jest. Cała tajemnica polega na tym, że nie można zapisać pliku Excela z zarejestrowanym makrem ze standardowym rozszerzeniem pliku.

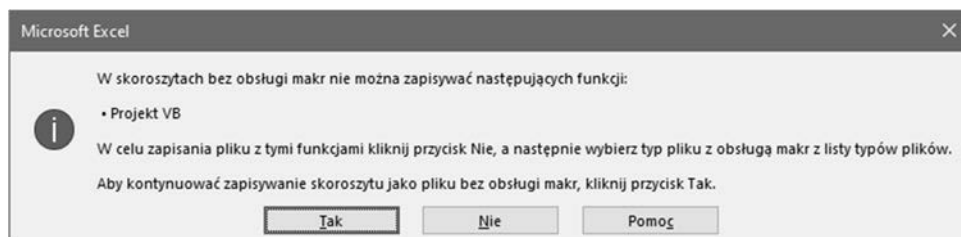
Przygotowania

Przy nadal otwartym pliku *02_RecordMacro.xlsx* zamknij edytor VBA.

Realizacja

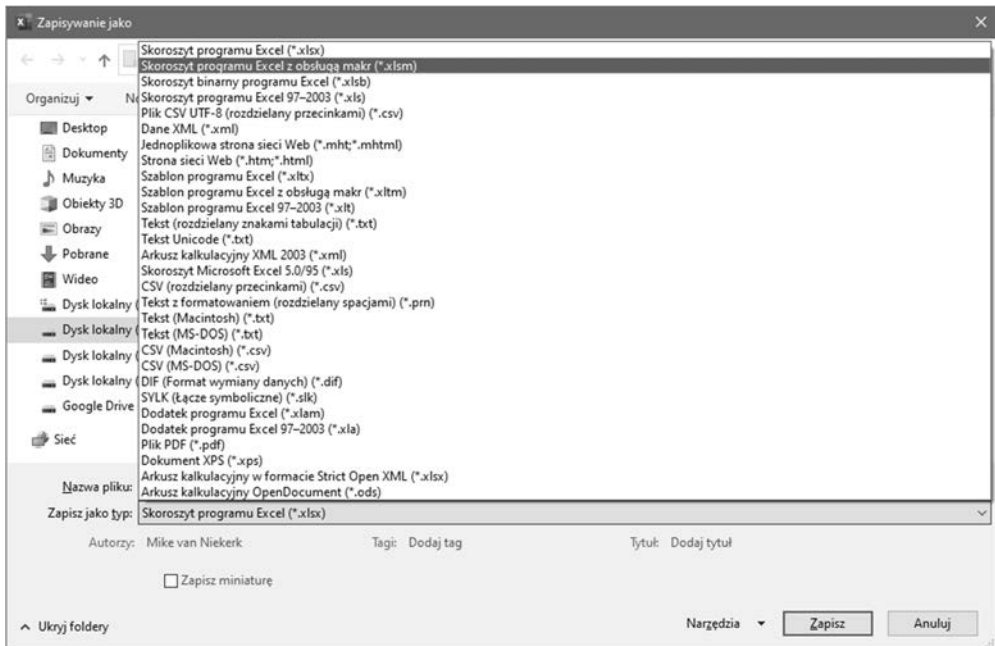
Wykonaj teraz następujące czynności:

1. W Excelu kliknij przycisk *Zapisz*. Pojawi się okno z ostrzeżeniem (zobacz rysunek 1.20).



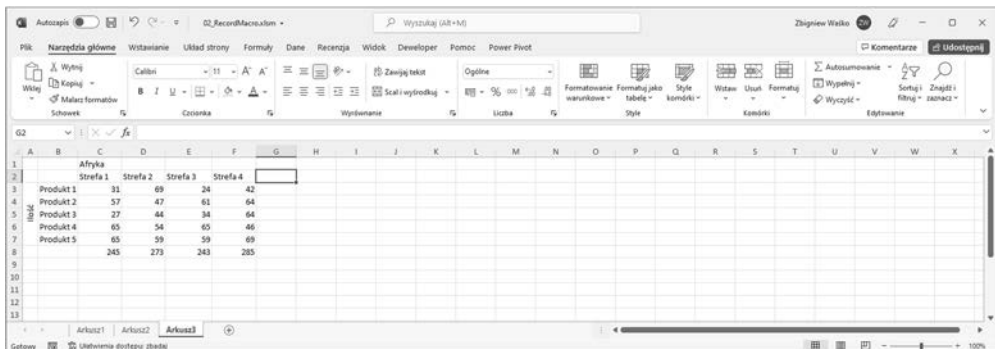
Rysunek 1.20. Okno z ostrzeżeniem

2. Ponieważ chcemy zapisać plik wraz z nowo zarejestrowanym makrem, kliknij przycisk *Nie*.
3. W oknie *Zapisywanie jako* wybierz lokalizację docelową dla pliku.
4. U dołu, poniżej pola *Nazwa pliku*, kliknij strzałkę w prawej części pola tekstowego *Zapisz jako typ*. Z wyświetlonej listy wybierz drugą pozycję od góry, *Skoroszyt programu Excel z obsługą makr (*.xlsm)* (zobacz rysunek 1.21).



Rysunek 1.21. Wybór pliku z obsługą makr w oknie dialogowym Zapisywanie jako

5. Kliknij przycisk *Zapisz*. Rozszerzenie pliku na pasku tytułowym zmieniło się na *.xlsm* (zobacz rysunek 1.22).



Rysunek 1.22. Zmodyfikowane rozszerzenie pliku widoczne na pasku tytułowym

Działanie

Warto tutaj zwrócić uwagę na dwie kwestie:

- Zapisanie makra w pliku ze standardowym rozszerzeniem *.xlsx* nie jest możliwe.
- Plik z wbudowanym makrem można zapisać jedynie z rozszerzeniem *.xlsm*.

Informacje dodatkowe

Na podstawie rozszerzenia pliku można stwierdzić, czy zawiera on makra.

Korzystanie z Centrum zaufania w kontekście bezpieczeństwa makr

Ten przepis dotyczy bezpieczeństwa, przynajmniej w odniesieniu do makr. Dowiesz się, gdzie w Excelu znajduje się Centrum zaufania i jak wybrać jeden z czterech poziomów zabezpieczeń makr.

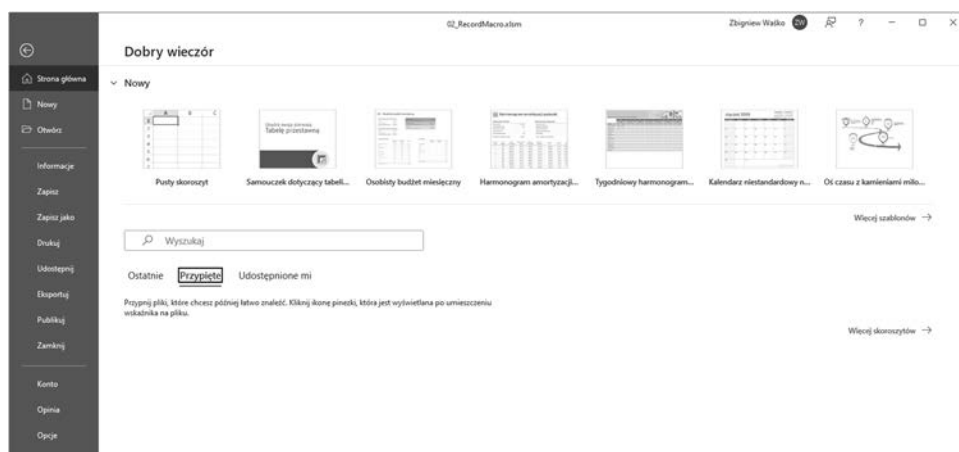
Przygotowania

Udało nam się już utworzyć i przetestować makro. Wprowadziliśmy nawet pewne zmiany w kodzie, modyfikując go w edytorze VBA. Ostatnim krokiem jest otwarcie pliku *02_RecordMacro.xlsm* w celu ustawienia zabezpieczeń makra.

Realizacja

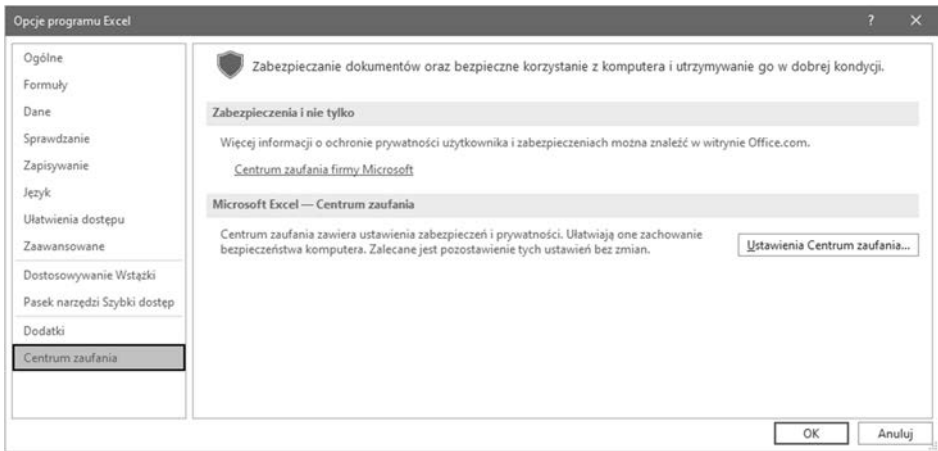
Wykonaj następujące kroki w ramach tego przepisu:

1. Kliknij zakładkę *Plik*, aby otworzyć widok *Backstage* (zobacz rysunek 1.23).



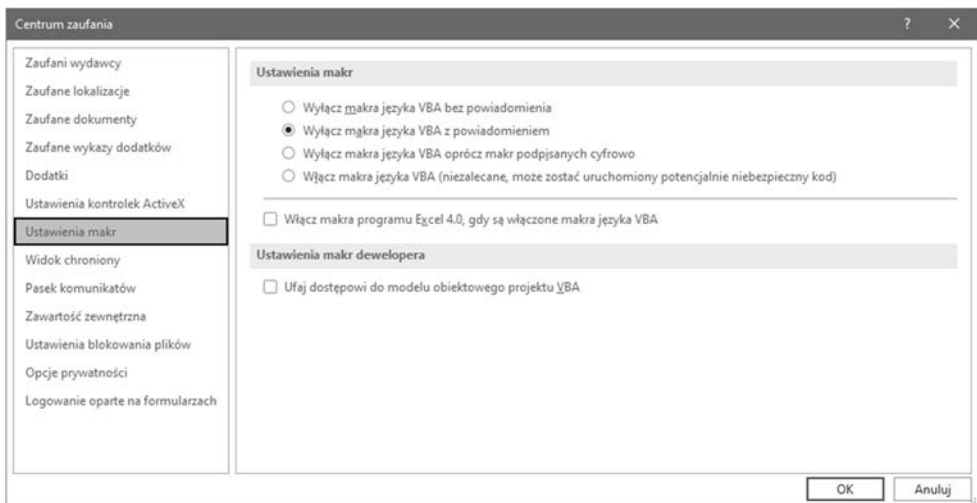
Rysunek 1.23. Widok Backstage

2. Z listy kategorii po lewej stronie wybierz ostatnią pozycję, *Opcje*. Otworzy się okno dialogowe *Opcje programu Excel* pokazane na rysunku 1.24.



Rysunek 1.24. Okno dialogowe Opcje programu Excel

3. Zaznacz ostatnią kategorię po lewej stronie, *Centrum zaufania*, a następnie po prawej stronie kliknij przycisk *Ustawienia Centrum zaufania*. Otworzy się okno pokazane na rysunku 1.25.



Rysunek 1.25. Ustawienia makr w oknie dialogowym Centrum zaufania

4. Po lewej stronie wybierz kategorię *Ustawienia makr*. W prawej części okna wyświetlą się szczegółowe informacje.
5. Włącz drugą od góry opcję, *Wyłącz makra języka VBA z powiadomieniem*.
6. Kliknij przycisk *OK*, aby zamknąć okno dialogowe *Centrum zaufania*.
7. W oknie dialogowym *Opcje programu Excel* kliknij *OK*.

Działanie

W *Ustawieniach makr* dostępne są cztery poziomy zabezpieczeń przed wirusami, które mogłyby zainfekować system. Do wyboru są następujące opcje:

- *Wyłącz wszystkie makra VBA bez powiadomienia.* Wybranie tej opcji spowoduje wyłączenie wszystkich makr. To ustawienie zapewnia bezpieczeństwo, ale czyni bezcelowym tworzenie makr.
- *Wyłącz wszystkie makra VBA z powiadomieniem.* Jest to ustawienie standardowe stosowane przez większość użytkowników. Po otwarciu pliku z osadzonym makrem zostanie wyświetlony żółty pasek komunikatu. Jeśli znasz i darzysz zaufaniem osobę, która wysłała plik, możesz śmiało włączyć makra. W przeciwnym razie pozostaw je wyłączone.
- *Wyłącz makra języka VBA oprócz makr podpisanych cyfrowo.* Uruchamiane będą tylko te makra, które mają podpis cyfrowy. Jednak w przypadku podpisów, które nie zostały oznaczone jako zaufane, nadal będzie wyświetlane ostrzeżenie o zabezpieczeniach.
- *Włącz makra języka VBA.* To ustawienie nie jest zalecane, ponieważ może zostać uruchomiony niebezpieczny kod. Nie wybieraj tej opcji, chyba że nie jesteś podłączony do internetu lub innych komputerów w sieci LAN.

Informacje dodatkowe

Kluczem jest czujność. W dzisiejszych czasach bezpieczeństwo komputera jest niezwykle ważne. Dobrym punktem wyjścia jest posiadanie programu antywirusowego, który będzie ciągle działał na komputerze. Lepiej się zabezpieczyć niż później żałować.

Tworzenie niestandardowej wstążki

Ten przepis ma charakter opcjonalny, ale może być bardzo przydatny.

Dostępnych jest kilka sposobów wywołania makra. Gdy już karta *Developer* jest dodana do wstążki, kliknięcie jej zakładki, a następnie przycisku *Makra* w grupie *Kod* nie stanowi większego problemu. A jeśli nie chcesz tego robić w taki sposób? Pomocne mogą być skróty klawiszowe, ale tylko do pewnego momentu, ponieważ makra potrafią się szybko mnożyć.

W tym przepisie pokażę, jak utworzyć nową kartę, grupę i ikony.

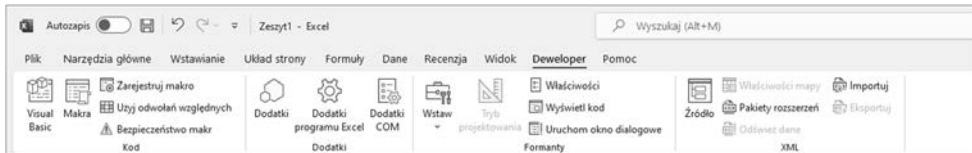
Przygotowania

Otwórz Excela z aktywnym nowym arkuszem¹.

¹ Aby zrealizować ten przepis w całości, należy otworzyć arkusz zawierający jakieś makra — *przypp. tłum.*

Realizacja

1. Zanim rozpoczniemy dostosowywanie programu, warto zapoznać się z prostym sposobem manipulowania wstążką Excela. Nie jest to tak naprawdę dostosowywanie, ale mimo to warto wiedzieć, jak kontrolować interfejs. Dwukrotne kliknięcie zakładki aktywnej karty powoduje zwinięcie wstążki. Ponowne dwukrotne kliknięcie zakładki rozwija kartę i wyświetla wstążkę w pełnej formie (zobacz rysunek 1.26).



Rysunek 1.26. Pełny widok wstążki Deweloper

Wstążka zwinięta jest pokazana na rysunku 1.27.

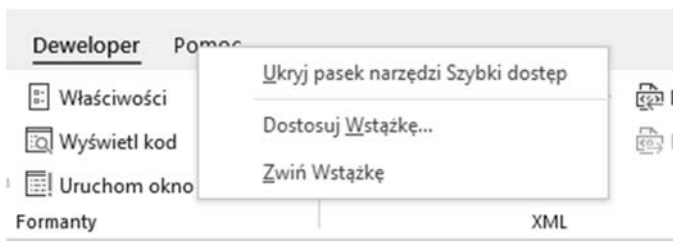


Rysunek 1.27. Wstążka zwinięta

Standardowe karty zazwyczaj zapewniają wszystkie potrzebne funkcje.

Jeśli jednak masz szereg funkcji, z których chcesz regularnie korzystać, a może też zbiór często używanych makr; bardzo pomocne będzie wyświetlanie skrótów do tych elementów na wstążce.

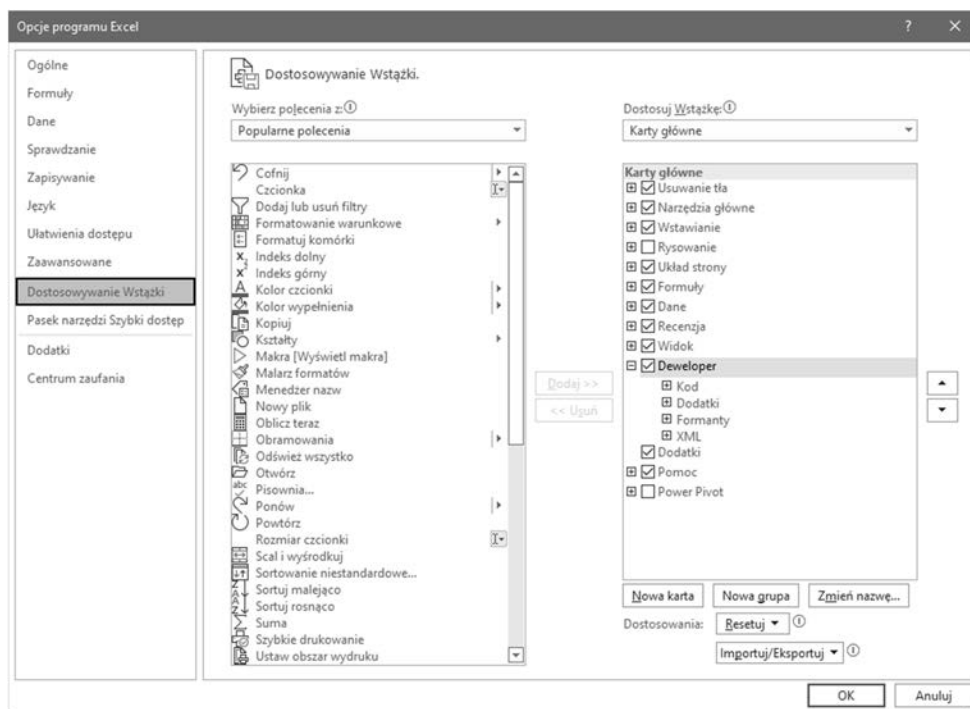
2. Aby dodać nową kartę do wstążki, kliknij prawym przyciskiem myszy dowolną istniejącą zakładkę i wybierz polecenie *Dostosuj Wstążkę* (zobacz rysunek 1.28).



Rysunek 1.28. Menu otwierane kliknięciem prawym przyciskiem myszy na zakładce Pomoc

3. Można również kliknąć prawym przyciskiem myszy w dowolnym miejscu na prawo od otwartej karty, w szarym obszarze, i wybrać polecenie *Dostosuj Wstążkę* albo wybrać *Plik/Opcje/Dostosowywanie Wstążki*.

Otworzy się okno dialogowe *Opcje programu Excel* z możliwością modyfikowania wstążki (zobacz rysunek 1.29).

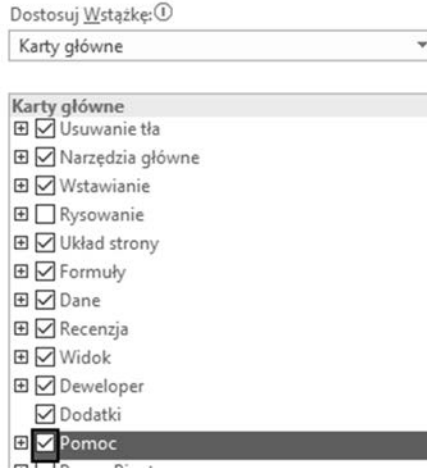


Rysunek 1.29. Okno dialogowe Opcje programu Excel

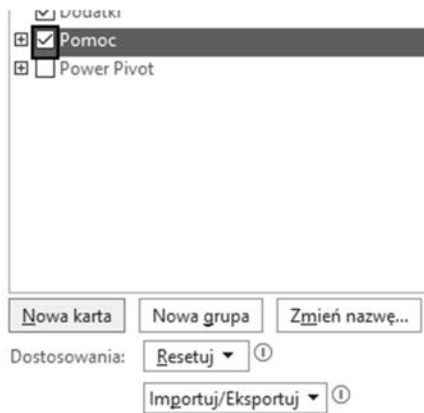
W oknie dialogowym *Opcje programu Excel* znajdują się dwie kolumny.

My zajmiemy się prawą kolumną z nagłówkiem *Dostosuj Wstążkę*.

4. Upewnij się, że wszystkie główne karty są zwinięte. Jeśli któraś z nich jest rozwinięta, kliknij przycisk – (minus), aby ją zwinać.
5. Zaznacz kartę *Pomoc*, ale jej nie rozwijaj, ponieważ chodzi tylko o to, aby nowa karta znalazła się po jej prawej stronie (zobacz rysunek 1.30).
6. Kliknij przycisk *Nowa karta* znajdujący się na dole kolumny (zobacz rysunek 1.31).
7. Na liście pojawi się nowa pozycja, o nazwie *Nowa karta (Niestandardowa)*, a pod nią *Nowa grupa (Niestandardowa)*, tak jak na rysunku 1.32.
8. Zaznacz pozycję *Nowa karta (Niestandardowa)*, a następnie kliknij przycisk *Zmień nazwę* znajdujący się pod listą kart (zobacz rysunek 1.33).
9. W oknie dialogowym *Zmianianie nazwy* wpisz właściwą nazwę nowej karty, potem kliknij przycisk *OK* (zobacz rysunek 1.34).
10. Zaznacz pozycję *Nowa grupa (Niestandardowa)*, jak na rysunku 1.35.



Rysunek 1.30. Zaznaczona karta Pomoc



Rysunek 1.31. Przycisk Nowa karta



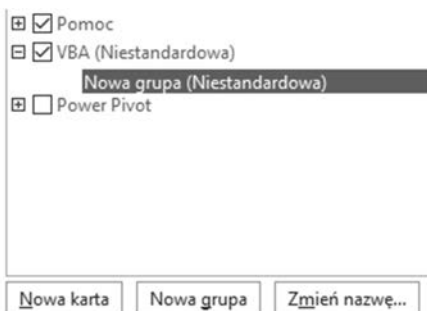
Rysunek 1.32. Wstawiona nowa karta



Rysunek 1.33. Zmiana nazwy nowej karty



Rysunek 1.34. Ustalanie nazwy nowej karty

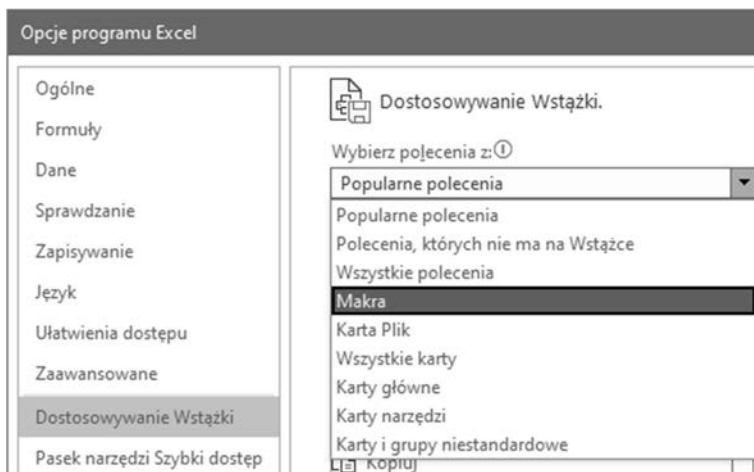


Rysunek 1.35. Zaznaczenie nowej grupy

11. Kliknij przycisk *Zmień nazwę*, a następnie wpisz nazwę nowej grupy. Kliknij przycisk *OK* (zobacz rysunek 1.36).
12. W lewej kolumnie wybierz zestaw poleceń o nazwie *Makra* (zobacz rysunek 1.37).
Pojawi się lista makr dostępnych w pliku, który jest otwarty (zobacz rysunek 1.38).
13. Zaznacz makro, które chcesz dodać do grupy, a następnie kliknij przycisk *Dodaj* (zobacz rysunek 1.39).



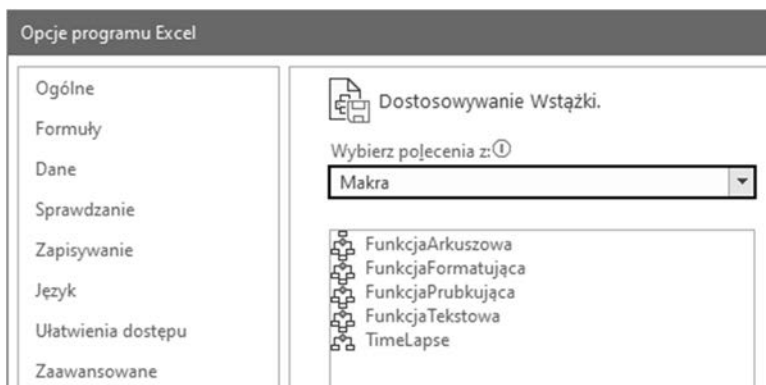
Rysunek 1.36. Okno dialogowe Zmianianie nazwy dla grup



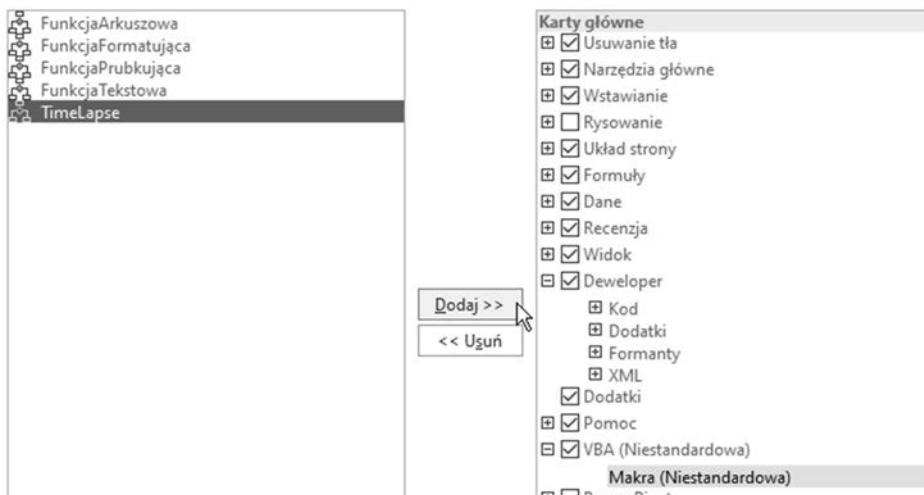
Rysunek 1.37. Wybór zestawu Makra

14. Zaznacz nowo wstawione polecenie i kliknij przycisk Zmień nazwę. W razie potrzeby wpisz bardziej opisową nazwę dla makra i wybierz pasującą do niego ikonę (zobacz rysunek 1.40). Kliknij OK, a następnie jeszcze raz OK.

Na wstążce pojawi się nowa karta z nową grupą i jedną ikoną w tej grupie (zobacz rysunek 1.41).



Rysunek 1.38. Lista zarejestrowanych makr



Rysunek 1.39. Kliknij przycisk Dodaj, aby dodać makro do grupy

15. Kliknij nową ikonę, aby uruchomić przypisane do niej makro. Możesz dodać więcej makr do tej samej grupy, powtarzając czynności od punktu 12. w tym przepisie.

Na podstawie powyższego przepisu można zapewnić sobie łatwy dostęp do wszystkich makr zarejestrowanych w skoroszybie.



Rysunek 1.40. Zmiana nazwy i ikony



Rysunek 1.41. Nowa karta z nową grupą

Skorowidz

A

analizowanie
 kodu, 24, 159
 przebiegu programu, 152
argument
 funkcji, 268
 wyświetlanie opisu, 274
arkusz kalkulacyjny
 wstawianie przycisku, 252
automatyczna kontrola składni, 184, 186
automatyczne wyświetlanie elementów, 84
automatyzacja, 168
autowypełnianie, 269

B

bezpieczeństwo makr, 40, 172
biblioteka obiektów
 PowerPointa, 289
 Worda, 277
błędy, 181
 danych, 198
 kompilacyjne, 182, 186
 logiczne, 198
 niewłaściwe odwołania, 198
 składniowe, 182–186
 skrajnych wartości, 198
 uruchomieniowe, 182, 187–193
 w kodzie, 185
 wersji, 199
 wyjątkowe, 199
 wznawianie działania procedury, 195

C

Centrum zaufania, 40
Charts (wykresy), 74

D

debuger, 204
debugowanie, 197
 metoda Debug.Print, 203
 okno
 komunikatu, 201
 Locals, 208
 Watches, 208
 techniki, 199, 203
 wykorzystanie kursora myszy, 199
definiowanie zasięgu zmiennych, 121
deklarowanie
 stałych, 116
 tablic, 124
 tablic dynamicznych, 127
 zmiennych, 109
dokowanie okien, 69
Dostosowywanie Wstążki, 25
Dostosuj Wstążkę, 44

E

edytor VBA, 34, 35, 50
 automatyczne formatowanie kodu, 68
 karta
 Docking, 70
 Editor, 69

edytor VBA
 Editor Format, 70
 General, 70
 z opcjami dokowania, 69
 z opcjami edytora, 67
 komponenty, 35
 lista obiektów, 80
 okno
 bezpośrednie, 57
 kodu, 53, 56
 projektu, 55
 właściwości, 55
 pasek
 menu, 54
 narzędziowy, 55
 tytułowy, 54
 przeglądarka obiektów, 82, 84
 tworzenie procedury, 59
 ustawienia ogólne, 69
 zmienianie ustawień, 66
 ekran powitalny, 51
 eksplorator projektu, 169
 etykieta TryAgain, 196
 Excel
 przenoszenie danych do PowerPointa, 293
 przenoszenie danych do Worda, 281

F

folder z szablonami plików, 285
 format wyświetlanych danych, 227
 formatowanie tekstu, 279
 formularz UserForm1, 231
 formularze
 kolory systemowe, 234
 kontrolki, 236, 246
 ustalanie kolejności, 259
 wycinanie, 243
 modyfikowanie właściwości, 232
 napis tytułowy, 234
 nazwa, 233
 ramki, 242
 testowanie, 249
 tworzenie, 89, 229
 widok w Excelu, 240
 wstawienie
 etykiety, 237
 pokrętła, 256
 pola kombo, 238, 254
 pola tekstowego, 238
 ramki, 243
 zamykanie, 250

formuła w arkuszu, 96
 funkcja
 Date (data), 264
 InputBox, 153, 220
 IntelliSense, 248, 280
 Intersect, 178
 JEŻELI, 269
 MsgBox, 204, 214, 219
 Now, 283
 Step Into, 118
 funkcje, 85, 141
 argumenty, 268
 arkuszowe, 145
 dodawanie opisu, 272
 niestandardowe, 262, 270, 272
 tworzenie, 88
 uruchamianie, 96
 wbudowane, 141
 własne, 147
 wyświetlanie opisu, 271
 wywoływanie, 98, 266

G

generowanie
 plików PowerPointa, 287
 prezentacji i slajdów, 290

H

hierarchia obiektów, 73

I

ikona
 Bezpieczeństwo makr, 172
 Label (etykieta), 236
 makra, 49
 Toggle Breakpoint, 206
 Wstaw funkcję, 272
 Zakładka, 284
 Zarejestruj makro, 62
 Zatrzymaj rejestrowanie, 30
 instancja
 PowerPointa, 290
 Worda, 276
 instrukcja
 .Selection.Paste, 283
 ElseIf, 155, 165
 End Sub, 86

GoTo, 152, 195
 Loop, 130
 On Error GoTo, 195
 On Error, 193
 ReDim, 127, 129
 Sub, 86
 Unload, 251
 Until, 166
 With, 282

K

karta

Deweloper, 24, 25
 Docking, 70
 Editor, 69
 Editor Format, 70
 General, 70
 Pomoc, 45
 z opcjami
 automatycznego formatowania kodu, 68
 dokowania, 69
 edytora, 67
 z ustawieniami ogólnymi, 69

kod

automatyczne formatowanie, 68
 kopiowanie i wklejanie, 65
 ręczne wprowadzanie, 64, 65
 wygenerowany z rejestracji makra, 62

kolekcja

Charts (wykresy), 74
 Sheets (arkusze), 74
 Workbooks (skoroszyty), 74
 Worksheets (arkusze kalkulacyjne), 74

kolekcje

modyfikowanie atrybutów, 74

kolory systemowe, 234

komentarze, 106

wstawione automatycznie, 108
 wstawione ręcznie, 108

komponenty edytora VBA, 35

konstrukcja

If-Then, 154
 Select Case, 157

kontrolka

pokrętło, 256
 pole kombo, 254
 Label (etykieta), 236
 TextBox (pole tekstowe), 237
 typu Przycisk, 92

kontrolki, 236

formularzy, 246
 ustalanie kolejności, 259
 wewnątrz ramki, 244
 właściwość Caption (napis), 236
 właściwość Name (nazwa), 238
 wyrównywanie, 242
 zaznaczanie, 241

L

lista

obiektów, 80
 parametrów okna, 215, 221, 224
 pola kombo, 255
 zarejestrowanych makr, 48
 zdarzeń, 174

Ł

łączenie ciągów znaków, 216

M

makra

bezpieczeństwo, 40, 172
 dodawanie do grupy, 48
 ikony, 49
 lista, 48
 odwołania względne, 102
 przypisywanie, 92
 rejestrowanie, 27, 62, 63, 101
 testowanie, 31
 uruchamianie, 90
 ustawienia, 41, 172
 zaawansowane, 99
 zapisywanie
 nowy skoroszyt, 104
 skoroszyt makr osobistych, 104
 ten skoroszyt, 104
 zmienianie nazwy, 49

makro, *Patrz także* procedura

formatowanie arkusza, 32
 mechanizm weryfikacji danych, 225

menu View (widok), 260

metoda, 78

Activate (aktywuj), 78, 134
 Application.InputBox, 223, 228
 ClearContents (wyczyść zawartość), 78
 Copy (kopiuj), 78, 137

- metoda
 - Debug.Print, 203
 - OnTime, 180
 - Select (zaznacz), 134, 137
 - metody obiektu Range, 136
 - model obiektowy, 71
 - moduł do przechowywania procedury, 58
 - modyfikowanie
 - atributów kolekcji, 74
 - okien dialogowych, 213
 - właściwości obiektów, 76
- ## N
- nagłówki w arkuszu, 110, 111
 - narzędzia
 - debugowania, 205
 - do tworzenia formularzy, 232
 - narzędzie
 - CommandButton, 245
 - Frame, 243
 - Select Objects, 241
 - Nowa grupa (Niestandardowa), 44
- ## O
- obiekt
 - Frame (ramka), 260
 - Range (zakres), 131
 - Workbook (skoroszyt), 80, 170
 - Zeszyt1, 74
 - obiekty, 72
 - hierarchia, 73
 - jako kolekcje, 74
 - lista, 80
 - metody, 78
 - modyfikowanie właściwości, 76
 - przeglądarka obiektów, 75, 82
 - obsługa
 - błędów, 181
 - błędu uruchomieniowego, 193
 - pokrętła, 256
 - poła kombo, 254
 - zdarzeń, 169
 - arkuszowych, 176
 - nieobiektowych, 178
 - skoroszytowych, 173, 174
 - odwołania
 - bezwzględne, 28, 100
 - do zakresów, 132, 134
 - względne, 28, 100
 - okna dialogowe
 - modyfikowanie, 213
 - tworzenie, 213
 - okno
 - Argumenty funkcji, 274
 - Centrum zaufania, 41, 172
 - edytora VBA, 26
 - Formularz wprowadzania nowej pozycji, 247
 - Immediate (okno bezpośrednie), 156, 159, 163, 202
 - komunikatu, 170, 215
 - lista parametrów, 215
 - łączenie ciągów znaków, 216
 - niestandardowe, 194
 - niewłaściwa odpowiedź, 219
 - przyciski Tak/Nie, 218
 - wyświetlanie danych, 217
 - wyświetlenie wartości zmiennej, 222
 - z datą, 143
 - z pozycją litery, 144
 - z wartością procentową, 143
 - z wynikiem, 149
 - Locals (zmiennych lokalnych), 112, 113, 146, 202
 - Makro, 26, 33, 60, 90
 - Opcje makra, 96
 - Opcje programu Excel, 25, 41
 - Options (opcje), 66, 67, 184
 - Properties (właściwości), 137, 232, 252
 - Properties w Excelu, 253
 - Przypisywanie makra, 92
 - References — VBAProject (odwołania — projekt VBA), 276, 277, 289
 - Rejestrowanie makra, 29, 63, 101
 - Tab Order, 260
 - Toolbox (przybornik), 236
 - UserForm1, 230
 - Watches (obserwacji), 208, 209
 - wprowadzania danych, 220
 - Application.InputBox, 223, 224
 - InputBox, 220
 - lista parametrów, 221, 224
 - wartość domyślna, 222
 - weryfikacja danych, 225
 - typ przyjmowanych danych, 227
 - Wstawianie funkcji, 272, 274
 - Zakładka, 284
 - Zmianianie nazwy dla grup, 47
 - opcja
 - Auto Syntax Check, 186
 - Cancel (anuluj), 174

- Debug (debuguj), 204
 - False (fałsz), 235
 - Function (funkcja), 264
 - Kontrolki ActiveX/Przycisk polecenia, 252
 - Makra, 34, 60
 - Module (moduł), 58, 64
 - Odwołania względne, 28
 - Public (publiczna), 89
 - Tab Order, 260
 - True (prawda), 235
 - UserForm (formularz użytkownika), 230
 - Użyj odwołań względnych, 100
 - View/Object Browser
(widok/przeglądarka obiektów), 83
 - Wyłącz makra języka VBA
z powiadomieniem, 172
 - opcje
 - edytora, 67
 - makra, 96
 - programu Excel, 25, 41
 - rejestrwania, 102
 - opis
 - argumentów funkcji, 271, 274
 - funkcji, 271, 272
 - ostrzeżenie, 183, 186
- P**
- paleta kolorów, 235
 - parametry
 - metody Application.InputBox, 224
 - okna komunikatu, 215
 - pasek narzędzi debugowania, 205
 - pętla, 161
 - For Each Next, 163
 - Do Loop Until, 162
 - Do Until, 162
 - For-Next, 163
 - pliki z obsługą makr, *.xlsm, 38, 39
 - pokrętło, 256
 - pola tekstowe PowerPointa, 291
 - pole kombo, 249, 254
 - testowanie, 256
 - pole tekstowe skrótu klawiszowego, 29
 - polecenie
 - Add Procedure (dodaj procedurę), 88, 263
 - Add Watch (dodaj obserwację), 209–211
 - Cut (wytnij), 243
 - Edytuj tekst, 94
 - Insert (wstaw), 230
 - Module (wstaw/moduł), 74, 87, 110, 179
 - Procedure (wstaw/procedurę), 88, 263
 - Paste (wklej), 244
 - Run/Run Sub/Userform, 89
 - Step Into (przejdź do), 112, 114, 156, 159
 - Tab Order (kolejność przełączania
klawiszem Tab), 260
 - Tools/Options (narzędzia/opcje), 67, 184
 - Tools/References (narzędzia/odwołania),
276, 288
 - vbNewLine, 216
 - View Code (widok kodu), 247
 - View/Locals Window (widok/okno
zmiennych lokalnych), 112, 126
 - View/Watch Window (widok/okno
obserwacji), 209
 - Wyczyść formaty, 63
 - Pomoc, 45
 - pomoc dla funkcji, 144
 - PowerPoint, 287
 - biblioteka obiektów, 289
 - generowanie prezentacji, 290
 - kopiowanie danych z Excela, 293
 - pola tekstowe, 291
 - tworzenie instancji, 288
 - procedura, 27, 85, 89, *Patrz także* makro
 - BeforeClose, 174
 - BeforePrint, 176
 - KopiowanieKomorek, 79
 - OdwołaniaWzględne, 102
 - PrzypisaniaKlawiszy, 180
 - TablicaDynamiczna, 127
 - TablicaStatyczna, 125
 - TworzenieDokumentuWorda, 278
 - TworzenieSlajdówPP, 293
 - Typy_danych, 119
 - Workbook_Open, 171
 - zdarzenia Open, 81
 - ZmianaCzcionki, 75
 - ZmianaWartosci, 77
 - procedury
 - bezpośrednie wykonywanie, 89
 - nazwa, 87
 - obsługi zdarzeń, 81, 169
 - prywatne, 89
 - publiczne, 89
 - tworzenie, 59, 86
 - wybór typu
 - Function (funkcja), 89
 - Property (właściwość), 89
 - Sub (procedura Sub), 85, 89
 - wykonywanie, 90, 91, 94
 - wznawianie działania, 195

programowanie obiektowe, 72
 przeglądarka obiektów, 82, 84
 menu kontekstowe, 83
 przybornik, 236
 przycisk
 Debug (debuguj), 187
 Move Down (przesuń w dół), 261
 Move Up (przesuń w górę), 261
 Nowa karta, 45
 Reset, 158
 Włącz zawartość, 172
 Zapisz nową pozycję, 250
 Zatrzymaj rejestrowanie, 101
 przyciski, 91, *Patrz także* kontrolki
 edycja nazwy, 94
 programowanie, 247
 przypisanie makra, 93
 zmienianie właściwości, 245
 punkt przerwania, 204
 wstawianie, 207

R

ramka z kontrolkami, 244
 Range, 133
 metody, 136
 odwołania, 133
 właściwości, 134
 Address (adres), 135
 Count (liczebność), 136
 Font (czcionka), 136
 Value (wartość), 135
 rejestrowanie makr, 27, 29, 62, 63, 101
 opcje, 102
 zaawansowanych, 99
 rozmiary danych, 121

S

Sheets (arkusze), 74
 skoroszyt makr osobistych, 104
 skrót klawiaturowy
 Alt + F11, 34, 110, 116
 Ctrl + A, 177
 Ctrl + C, 64
 Ctrl + G, 57, 156
 Ctrl + P, 175
 Ctrl + Q, 104
 Ctrl + Shift + C, 104
 Ctrl + Shift + F, 28
 Ctrl + Shift + Q, 107

Ctrl + Shift + R, 101
 Ctrl + spacja, 248
 Ctrl + V, 64, 244
 Ctrl + X, 243
 Ctrl + Z, 179
 F5, 89
 F7, 52
 F8, 112, 266
 F9, 208
 przypisany do makra, 94
 Shift + Ctrl + strzałka w prawo, 179
 slajdy, 290
 tworzenie, 293
 stałe, 116
 szablony, 284

T

tablice, 124
 dynamiczne, 127, 129
 jednowymiarowe, 128
 wielowymiarowe, 126, 129
 testowanie, 189
 formularza, 249
 makra, 31
 pola kombo, 256
 tryb projektowania, 253
 tworzenie
 dokumentów Worda, 275–278
 formularzy, 229
 funkcji, 86, 147
 funkcji niestandardowych, 262
 instancji PowerPointa, 288
 kodu VBA, 61
 modułu, 58
 niestandardowej wstążki, 42
 nowej grupy, 46
 okien dialogowych, 213
 procedury, 59, 86
 przycisków, 91
 skrótów klawiszowego, 96
 slajdów, 293
 typy danych, 119, 121, 227

U

UDF, User-Defined Function, 86
 uruchamianie
 funkcji, 96
 makr, 90
 ustawienia makr, 41, 172

usuwanie

- formatowania komórek, 177
- wierszy kodu, 35

V

VBA, Visual Basic for Applications, 23

- debugowanie, 197
- formularze, 229
- funkcje wbudowane, 141
- komentarze, 106
- model obiektowy, 71
- obiekt Range, 131
- obsługa
 - błędów, 181
 - PowerPointa, 287
 - Worda, 275
- okna dialogowe, 213
- stałe, 116
- tablice, 124
- typy danych, 119
- zmienne, 109

W

weryfikacja danych, 225

widok

- Backstage, 24, 40
- chroniony, 24
- wstążki Deweloper, 43

właściwości

- formularza, 232
- obiektów, 76
- obiektu Range, 134
- pokrętła, 257
- przycisku, 245

właściwość

- Address (adres), 135, 178
- Caption (napis), 233, 236, 240, 252
- Count (liczebność), 136
- Font (czcionka), 136
- Name (nazwa), 137, 233
- RowSource (źródło danych), 255

Word

- formatowanie tekstu, 279, 280
- kopiowanie danych z Excela, 281
- szablony, 284
- tworzenie, 275–278
- wpisywanie tekstu, 279
- zakładki, 284
- zamknięcie dokumentu, 283

Workbook (skoroszyt), 80

Workbooks (skoroszyty), 74, 134

Worksheets (arkusze kalkulacyjne), 74, 134

wstawianie przycisku do arkusza, 252

wstążka

- Deweloper, 43
- niestandardowa, 42

wykonywanie

- funkcji
 - formuła, 96
 - wywołanie z procedury, 98
- procedur, 89
- klawisz skrótu, 94
- okno Makro, 90
- przyciski, 91

wyrównywanie kontroltek, 242

wyszukiwanie błędów, 189

wyświetlanie

- danych, 227
- kodu, 34
- opisu argumentu, 274
- opisu funkcji, 271

wywoływanie funkcji, 266

Z

zakładki, 284

- nazwa, 285

zakresy, 30, 131, *Patrz także* Range

- nazwy, 133
- zastosowanie, 138
- zaznaczone, 30, 133

zamykanie formularza, 250

zapisywanie

- makr, 104
- pliku, 38

Zarejestruj makro, 62

zasięg zmiennych, 121

Zatrzymaj rejestrowanie, 30

zaufane dokumenty, 173

zaznaczenie

- kontroltek, 241
- obiektów, 241
- zakresów, 30, 133

zdarzenia

- arkuszowe, 176
- lista, 174
- nieobiektywne, 178
- procedury obsługi, 169
- skoroszytowe, 173

zdarzenie

- BeforeClose, 175
- BeforeSave, 176
- Change, 177
- Click, 247
- Initialize, 258
- OnKey, 179
- Open, 81
- SelectionChange, 178

zmiana

- odniesienia do komórki, 37
- właściwości przycisku, 245

zmienna

typu

- Integer, 120
- Long, 120
- Variant, 120

wdApp, 278

zmiennie, 109

- lokalne, 122
- modułowe, 123
- obserwowane w oknie Watches, 210, 212
- publiczne, 123
- zasięg, 121

znak

- &, 216
- apostrofu ('), 107
- równości (=), 97

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 i Excel: przekonaj się, jak łatwe jest programowanie!

Visual Basic for Applications (VBA) jest dobrze znanym językiem programowania; w pakiecie Microsoft Office zaimplementowano jego siódmą wersję. Skrypty VBA świetnie się sprawdzają w automatyzowaniu pracy w aplikacjach MS Office, pozwalają też na tworzenie własnych funkcji zwiększających możliwości tych programów. Co więcej, VBA jest językiem, który wyjątkowo łatwo opanować. Nawet te osoby, które nic nie wiedzą o programowaniu, błyskawicznie ucą się pisania kodu VBA i szybko mogą się cieszyć efektami swojej pracy.

Dzięki tej książce nauczysz się programować w języku VBA. Dowiesz się także, w jaki sposób można zautomatyzować wiele żmudnych czynności wykonywanych w Excelu i w innych aplikacjach pakietu MS Office. Naukę rozpoczniesz od podstaw, które podano tu w przystępnej i zrozumiałej formie. Nauczysz się nagrywać makra, a później stopniowo będziesz przechodzić do bardziej złożonych zadań programistycznych. Zapoznasz się też z zaawansowanymi narzędziami wbudowanymi w Excela i przekonasz się, jak bardzo mogą one zwiększyć Twoją produktywność. Dodatkowym plusem jest mnóstwo ćwiczeń, wskazówek i propozycji praktycznych projektów z wykorzystaniem Excela, Worda i PowerPointa. Pracę ułatwią Ci również przykładowe kody i filmy instruktażowe. Korzystaj do woli!

W książce między innymi:

- język VBA w środowisku aplikacji pakietu MS Office
- tworzenie własnych procedur, funkcji i formularzy w VBA
- operowanie na arkuszach Excela
- wykrywanie i obsługa błędów
- tworzenie własnych okien dialogowych i pobieranie danych użytkownika
- poszerzanie możliwości aplikacji z pakietu Office

Mike Van Niekerk jest wykładowcą i certyfikowanym instruktorem Microsoftu z ogromnym doświadczeniem w posługiwaniu się aplikacjami MS Office. Prowadzi szkolenia na wszystkich poziomach. Jest także autorem podręczników z zakresu obsługi pakietu MS Office, przeznaczonych dla kilku firm szkoleniowych z branży IT w RPA.

| | | |
|--|--|---|
|  | KOD KORZYŚCI Sięgnij po więcej! ▶ |  |
|  helion.pl | ISBN 978-83-283-9631-9 | |
|  HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl |  9 788328 396319 | |
| Cena: 79,00 zł | | |

Packt