

Piotr Janus

VBA na sto dwa,

czyli 102 ćwiczenia z wykorzystaniem VBA



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek 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 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.

Redaktor prowadzący: Szymon Szwajger
Projekt okładki: Studio Gravite / Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

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/vba100>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-7073-9

Copyright © Helion SA 2021

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Wstęp	7
1. Tabliczka mnożenia	11
2. Rok przestępny	17
3. Piątek trzynastego	19
4. Kalkulator wieku	25
5. Miesiące na literę	29
6. Dni wolne od pracy	33
7. Liczby palindromiczne	37
8. NWD i NWW	39
9. Liczby pierwsze	43
10. Liczby półpierwsze	47
11. Liczby emirp	49
12. Liczby pierwsze bliźniacze	53
13. Tabliczka mnożenia — cd.	55
14. Trójkąt Floyda	59
15. Trójkąt Pascala	61
16. Szyfr gaderypoluki	65
17. Szyfr zegarowy	69
18. Alfabet Morse’a	73

19. Liczby narcystyczne	77
20. Liczby automorficzne	81
21. Liczby kanadyjskie	83
22. Liczby faktorion	87
23. Liczby Nivena	89
24. Wartości unikatowe i duplikaty — komórki	91
25. Sortowanie arkuszy	95
26. Sortowanie cyfr w liczbach	97
27. Porównywanie liczb	101
28. Zapełnianie komórek	103
29. Piwotowanie danych	107
30. Sortowanie elementów tablicy	111
31. Elementy unikatowe i wspólne — tablice	115
32. Liczby doskonałe i prawie doskonałe	121
33. Liczby zaprzyjaźnione	125
34. Liczby Sastry'ego	129
35. Liczby Münchhausena	131
36. Liczby Catalana	133
37. Zygodromy	135
38. Czas zimowy, czas letni	137
39. Palindromy	141
40. PESEL	143
41. Koniunkcja planet	149
42. Miesiąc synodyczny	151
43. Wielkanoc	155
44. Generator liczb losowych	159
45. Wskaźnik BMI	165
46. Liczby rzymskie	169

47. Ciąg Fibonacciego	175
48. Ciąg Jacobsthala	179
49. Liczby wielokątne	181
50. Podział koła	185
51. Suma magiczna i magiczny kwadrat	187
52. Ciąg Conwaya	191
53. Kombinatoryka	193
54. Sto drzewi	201
55. Przelicznik walut	203
56. Odległość na kuli	207
57. Ślimak	211
58. Zygzak	215
59. Liczba kolista	219
60. Liczby taksówkowe	221
61. Liczba wampir — policz kły	225
62. Zamiana godzin na tekst	229
63. Zamiana liczb na tekst	235
64. Szyfr Cezara	241
65. Szyfr Vigenere’a	245

Wstęp

Drogi Czytelniku!

Książka *VBA na sto dwa, czyli 102 ćwiczenia z wykorzystaniem VBA* ma za zadanie sprawdzić, ugruntować i poszerzyć Twoją wiedzę na temat Visual Basic for Applications. Zakładam, że zapoznałeś się już z teorią i chcesz ją wykorzystać w praktyce. Zapewne Twoim celem jest zacząć tworzyć własne programy o praktycznym zastosowaniu, ale czujesz, że jeszcze na to za wcześnie. Ta książka może Ci pomóc osiągnąć ten cel.

Rozwiązanie zaproponowanych ćwiczeń wymaga stosowania pętli For-Next, For Each-Next, Do-Until, Do-While, struktur If-Then-Else, Select Case, funkcji Excela i VBA, odwołania do metod lub właściwości obiektów itd. Wielokrotne wykonanie podobnych czynności gwarantuje, że polecenia zapadną w pamięć. Książka pomaga wykształcić programistyczny sposób myślenia niezbędny do tworzenia własnych rozwiązań.

Każdy z rozdziałów składa się z jednego lub kilku ćwiczeń, które polegają na napisaniu procedur Sub lub Function. Dla uproszczenia w dalszej części książki procedury Sub będzie nazywać się procedurami, a procedury Function — funkcjami. Do każdego polecenia dołączone jest przykładowe rozwiązanie. Pamiętaj jednak, że jest wiele dróg do celu! Zamieszczone w książce rozwiązanie to tylko propozycja.

Jestem przekonany, że *VBA na sto dwa, czyli 102 ćwiczenia z wykorzystaniem VBA* pomoże Ci rozwinąć się jako programista Visual Basic for Applications. Jednocześnie mam nadzieję, że rozwiązywanie zadań będzie dla Ciebie ciekawą intelektualną przygodą i mile spędzonym czasem. Do dzieła!

Oznaczenia

Większy stopień trudności (✖)

Część ćwiczeń opatrzone gwiazdką. Oznacza to, że są one nieco trudniejsze. Pamiętaj jednak, że ocena stopnia skomplikowania ćwiczenia jest czysto subiektywna. Ćwiczenia z gwiazdką mogą okazać się dla Ciebie łatwe, natomiast te teoretycznie łatwiejsze — wręcz przeciwnie. Nie sugeruj się więc przesadnie oznaczeniem, a przede wszystkim nie daj się zniechęcić do podjęcia wyzwania.

Komentarz (💬)

Komentarz zawierający wyjaśnienie zastosowanego w danym przypadku rozwiązania, omawiający funkcję Excel lub VBA, zwracający uwagę na jakiś szczegół lub sugerujący dobrą praktykę.

Czas działania programu (🕒)

Pomiar czasu działania procedury lub funkcji. Porównaj ze swoim pomiarem.

Czas pomiaru to średnia z trzech pomiarów, ponieważ poszczególne z nich mogą się od siebie różnić.

Pamiętaj, że czas działania programu zależy od parametrów komputera, jakim dysponujesz. Pomiarów dokonano na dwóch komputerach:

- ♦ pomiar po lewej: ASUS X550CC, procesor i3-3217U 1.80 Ghz, 8 GB RAM. System operacyjny Windows 10 64 bity, MS Excel 2016 PL 64 bity.
- ♦ pomiar po prawej: ASUS X556UQK, procesor i5-7200 2.50 Ghz, 8 GB RAM. System operacyjny Windows 10 64 bity, MS Excel 2013 PL 32 bity.

Jeżeli masz szybszy komputer, a pomiar czasu gorszy od zamieszczonego w książce, prawdopodobnie Twój kod wymaga jeszcze usprawnień.

Aby dokonać pomiaru w sekundach, skorzystaj z następującego kodu:

```
Sub CountTime()  
    Dim Start As Double  
    Dim Finish As Double
```

```
Start = Timer  
    'tu umieść kod  
Finish = Round(Timer - Start, 2)  
MsgBox Finish & " sekund"  
End Sub
```

Ciąg liczb całkowitych (OEIS)

OEIS (*On-Line Encyclopedia of Integer Sequences*) to internetowa baza ciągów liczb całkowitych. Jest skierowana zarówno do naukowców, jak i amatorów matematyki. Każdy wpis zawiera początkowe elementy ciągu, słowa kluczowe, przypisy itp. Bazę można przeszukiwać w oparciu o ciąg, słowo (po angielsku) lub numer ciągu.

Symbol oznacza, że celem zadania jest stworzenie ciągu, który znajduje się w bazie OEIS pod wskazanym indeksem. Wpisz wartość indeksu w wyszukiwarce bazy, aby dowiedzieć się więcej na temat interesującego Cię ciągu.

1. Tabliczka mnożenia

Ćwiczenie

- 1** Napisz procedurę, która utworzy tabliczkę mnożenia od 1 do 100, jak na poniższym obrazku.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Rozwiązania

Option Explicit

```
Sub MultiplicationTable()  
    Dim Row As Byte, Col As Byte  
    Dim Num As Byte  
    Dim Sht As Worksheet  
    Set Sht = ThisWorkbook.Worksheets(1)  
    For Row = 1 To 10  
        For Col = 1 To 10  
            Num = Row * Col  
            Sht.Cells(Row, Col).Value = Num  
        Next Col  
    Next Row  
End Sub
```

```
Sub MultiplicationTable()  
    Dim myArray(1 To 10) As Byte
```

```
Dim Row As Byte, Col As Byte
Dim Num As Byte
Dim Sht As Worksheet
Set Sht = ThisWorkbook.Worksheets(1)
For Row = LBound(myArray) To UBound(myArray)
    For Col = LBound(myArray) To UBound(myArray)
        Num = Row * Col
        Sht.Cells(Row, Col).Value = Num
    Next Col
Next Row
End Sub
```



W VBA należy mieć na uwadze, że ten sam cel można osiągnąć na wiele różnych sposobów. W drugim rozwiązaniu tabliczka mnożenia została utworzona z wykorzystaniem tablicy.



W jaki sposób zachować ciągłą kontrolę nad kodem, który piszemy?

1. Uruchom okienko *Immediate* (skrót klawiszowy *Ctrl+G*) i umieść w kodzie `Debug.Print Num`. Stosuj tak często, jak potrzebujesz. Śledź w okienku *Immediate* pojawiające się wartości.

```
Sub MultiplicationTable()
    Dim Row As Byte, Col As Byte
    Dim Num As Byte
    Dim Sht As Worksheet
    Set Sht = ThisWorkbook.Worksheets(1)
    For Row = 1 To 10
        For Col = 1 To 10
            Num = Row * Col
            Debug.Print Num
            Sht.Cells(Row, Col).Value = Num
        Next Col
    Next Row
End Sub
```

Immediate

7
8
9
10
2

2. Dla małych pętli stosuj przycisk *F8*, aby krok po kroku przechodzić przez program.

```
Sub MultiplicationTable()  
    Dim Row As Byte, Col As Byte  
    Dim Num As Byte  
    Dim Sht As Worksheet  
    Set Sht = ThisWorkbook.Worksheets(1)  
    For Row = 1 To 10  
        For Col = 1 To 10  
            Num = Row * Col  
            Sht.Cells(Row, Col).Value = Num  
        Next Col  
    Next Row  
End Sub
```

```
Sub MultiplicationTable()  
    Dim Row As Byte, Col As Byte  
    Dim Num As Byte  
    Dim Sht As Worksheet  
    Set Sht = ThisWorkbook.Worksheets(1)  
    For Row = 1 To 10  
        For Col = 1 To 10  
            Num = Row * Col  
            Sht.Cells(Row, Col).Value = Num  
        Next Col  
    Next Row  
End Sub
```

3. Dla większych pętli stosuj breakpointy. Zaznacz wiersz i wciśnij *F9* lub kliknij kursorem szare pole na wysokości wiersza, gdzie chcesz umieścić breakpoint.

```
Sub MultiplicationTable()  
    Dim Row As Byte, Col As Byte  
    Dim Num As Byte  
    Dim Sht As Worksheet  
    Set Sht = ThisWorkbook.Worksheets(1)  
    For Row = 1 To 10  
        For Col = 1 To 10  
            Num = Row * Col  
            Sht.Cells(Row, Col).Value = Num  
        Next Col  
    Next Row  
End Sub
```

Procedura uruchomiona przyciskiem *F5* zatrzyma się na breakpointcie. Wielokrotnie przyciskaj *F5*, aby przechodzić przez program.

```
Sub MultiplicationTable()
    Dim Row As Byte, Col As Byte
    Dim Num As Byte
    Dim Sht As Worksheet
    Set Sht = ThisWorkbook.Worksheets(1)
    For Row = 1 To 10
        For Col = 1 To 10
            Num = Row * Col
            Sht.Cells(Row, Col).Value = Num
        Next Col
    Next Row
End Sub
```

Pamiętaj, że jeżeli stosujesz breakpoints lub krokowo przechodzisz przez kod, w każdej chwili masz możliwość podejrzenia wartości zmiennej po najechaniu na nią kursorem.

```
Sub MultiplicationTable()
    Dim Row As Byte, Col As Byte
    Dim Num As Byte
    Dim Sht As Worksheet
    Set Sht = ThisWorkbook.Worksheets(1)
    For Row = 1 To 10
        For Col = 1 To 10
            Num = Row * Col
            Sht.Cells(Row, Col).Value = Num
        Next Col
    Next Row
End Sub
```

Wyczyść wszystkie breakpoints za pomocą skrótu klawiszowego *Ctrl+Shift+F9*.

Ćwiczenie

2

Napisz procedurę, która utworzy tabliczkę mnożenia dla liczb nieparzystych od 1 do 81, jak na poniższym obrazku.

	A	B	C	D	E
1	1	3	5	7	9
2	3	9	15	21	27
3	5	15	25	35	45
4	7	21	35	49	63
5	9	27	45	63	81

Rozwiązania

```
Option Explicit

Sub MultiplicationTableTwoStep()
    Dim Row As Byte, Col As Byte
    Dim i As Byte, j As Byte
    Dim Num As Byte
    Dim Sht As Worksheet
    Set Sht = ThisWorkbook.Worksheets(1)
    Row = 1
    For i = 1 To 10 Step 2
        Col = 1
        For j = 1 To 10 Step 2
            Num = i * j
            Sht.Cells(Row, Col).Value = Num
            Col = Col + 1
        Next j
        Row = Row + 1
    Next i
End Sub
```



Instrukcja `Option Explicit` używana jest na poziomie modułu i wymusza deklarowanie wszystkich zmiennych w tym module. W przypadku próby użycia niezadeklarowanej zmiennej nastąpi błąd podczas kompilacji.

Zaletą stosowania `Option Explicit` jest to, że chroni przed błędami powstającymi na skutek niepoprawnego wpisania nazwy zmiennej. Literówka w nazwie zmiennej sprawi, że Visual Basic potraktuje ją jako nową zmienną i zażąda jej zadeklarowania.

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 

Co Ty wiesz o VBA?

Co wiesz i co potrafisz — to dwie różne sprawy. Od wiedzy do umiejętności prowadzi droga usłana... praktyką. A praktykę najlepiej i najbezpieczniej zdobywa się nie w zawodowym boju, gdy na ręce patrzą szef, koledzy z teamu i klient na dodatek, tylko w zaciszu własnego pokoju, przy ulubionym biurku, na ulubionym krześle, w czasie wolnym. Ćwicząc i bawiąc się po prostu! I właśnie w tym celu powstała ta książka. Autor zawarł w niej 102 ćwiczenia, dzięki którym przejdiesz od teorii do praktyki Visual Basic for Applications i będziesz gotów do swobodnego tworzenia własnych programów.

Rozwiązanie zaproponowanych ćwiczeń wymaga stosowania pętli For...Next, For Each...Next, Do...Until, Do...While, struktur If...Then...Else, Select Case, funkcji Excel i VBA, odwołania do metod lub właściwości obiektów itd. Wielokrotne wykonywanie podobnych czynności gwarantuje, że polecenia zapadną Ci w pamięć. Książka pomoże Ci wykształcić programistyczny sposób myślenia, niezbędny do tworzenia własnych rozwiązań. Każdy rozdział składa się z jednego lub kilku ćwiczeń, które polegają na napisaniu procedur Sub lub Function. Do wszystkich poleceń dołączono przykładowe rozwiązania. Na początek możesz z nich skorzystać, potem jednak szukaj własnych rozwiązań.

Do dzieła:

- **Stwórz tabliczkę mnożenia**
- **Sprawdź, kiedy wypadnie najbliższy piątek trzynastego**
- **Policz dni wolne od pracy w kolejnym roku**
- **Napisz własny przelicznik walut**
- **Dowiedz się, czy utrzymujesz prawidłową masę ciała...**

...a to dopiero początek zabawy!

Helion 	<i>Sprawdź nasze szkolenia!</i> SZKOLENIA  AKADEMIA IT & BUSINESS HELIONSZKOLENIA.PL	KOD KORZYŚCI <i>Sięgnij po więcej!</i>  
 helion.pl  HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl		ISBN 978-83-283-7073-9  9 788328 370739
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 49,00 zł