

Excel

z a a w a n s o w a n y

AUTOMATYZACJA PRACY Z UŻYCIEM MAKR

NPV
WSPKORELACJI
ROZKŁ.EXP
KOMÓRKA
VBA
LOG
SUMA
CZY.LICZBA
JEŻELI
COS
DNI.ROBOCZE
ILOCZYN

Tom XII



Automatyzacja pracy z użyciem makr

Piotr Dynia,
Jakub Kudliński



Autorzy:

Piotr Dynia, Jakub Kudliński

Kierownik grupy wydawniczej:

Ewa Ziętek-Maciejczyk

Wydawca:

Monika Kijok

Redaktor prowadzący:

Rafał Janus

Korekta:

Zespół

Skład i łamanie:

Norbert Bogajczyk

Projekt okładki:

Piotr Fedorczyk

Druk: Miller

ISBN: 978-83-269-3772-9

Copyright by Wydawnictwo Wiedza i Praktyka sp. z o.o.

Warszawa 2015

Wydawnictwo Wiedza i Praktyka sp. z o.o.

03-918 Warszawa, ul. Łotewska 9a

tel. 22 518 29 29, faks 22 617 60 10

NIP: 526-19-92-256

Numer KRS: 0000098264 – Sąd Rejonowy dla m.st. Warszawy, Sąd Gospodarczy

XIII Wydział Gospodarczy Rejestrowy. Wysokość kapitału zakładowego: 200.000 zł

„Automatyzacja pracy z użyciem makr” wraz z przysługującymi Czytelnikom innymi elementami dostępnymi w subskrypcji (e-letter, strona WWW i inne) chronione są prawem autorskim. Przedruk materiałów opublikowanych w książce „Automatyzacja pracy z użyciem makr” oraz w innych dostępnych elementach subskrypcji – bez zgody wydawcy – jest zabroniony. Zakaz nie dotyczy cytowania publikacji z powołaniem się na źródło.

Publikacja „Automatyzacja pracy z użyciem makr” została przygotowana z zachowaniem najwyższej staranności i wykorzystaniem wysokich kwalifikacji, wiedzy i doświadczenia autorów oraz konsultantów. Zaproponowane w publikacji „Automatyzacja pracy z użyciem makr” oraz w innych dostępnych elementach subskrypcji wskazówki, porady i interpretacje nie mają charakteru porady prawnej. Ich zastosowanie w konkretnym przypadku może wymagać dodatkowych, pogłębionych konsultacji. Publikowane rozwiązania nie mogą być traktowane jako oficjalne stanowisko organów i urzędów państwowych. W związku z powyższym redakcja nie może ponosić odpowiedzialności prawnej za zastosowanie zawartych w publikacji „Automatyzacja pracy z użyciem makr” lub w innych dostępnych elementach subskrypcji wskazówek, przykładów, informacji itp. do konkretnych przypadków.

Spis treści

1. Informacje wstępne	7
1.1. Przygotowanie własnej procedury	8
1.2. Instrukcja MsgBox – wyświetlanie informacji w oknie	10
1.3. Sposoby uruchamiania makra	12
2. Zmienne i operacje arytmetyczne	14
2.1. Wynik dodawania w oknie dialogowym	14
2.2. Kolejność wykonywania operacji arytmetycznych	15
2.3. Zmienne w języku VBA	16
2.4. Działania arytmetyczne za pomocą zmiennych	17
2.5. Deklarowanie zmiennych	18
3. Typy zmiennych i operacje na tekstach	23
3.1. Określanie typu zmiennej w deklaracji	23
3.2. Łączenie ze sobą dwóch tekstów	25
3.3. Odczytywanie fragmentu tekstu	27
4. Operacje na tekście i instrukcja warunkowa <i>If</i>	30
4.1. Pobranie informacji od użytkownika za pomocą okna dialogowego	30
4.2. Wyszukiwanie określonego tekstu w innym tekście	31
4.3. Instrukcja <i>if</i> – tworzenie bloków warunkowych	33
4.4. Przydatne funkcje tekstowe	35
5. Operacje na datach i wartościach czasu, rozszerzona instrukcja <i>MsgBox</i>	36
5.1. Przechowywanie dat i wartości czasu	36
5.2. Własny format daty	38
5.3. Przydatne funkcje operujące na datach	39
5.4. Rozszerzone możliwości instrukcji <i>MsgBox</i>	40
6. Obiekty, metody i właściwości	44
6.1. Właściwości obiektu	44
6.2. Metody obiektu	45
6.3. Operacje na całym skoroszycie	47
6.4. Kolekcja – operowanie na większej liczbie obiektów	49
7. Obiekt <i>Worksheet</i>, pętla po elementach kolekcji i instrukcja <i>Select Case</i>	51
7.1. Operacje na arkuszach	51
7.2. Zmienne obiektowe	53

7.3. Operacje na kolekcji obiektów	53
7.4. Instrukcje warunkowe tworzone za pomocą instrukcji Select Case	55
8. Obiekt <i>Range</i>, instrukcja <i>With</i> i formatowanie komórek	58
8.1. Operacje na komórkach arkusza	58
8.2. Formatowanie komórek	60
8.3. Upraszczenie kodu – zastosowanie instrukcji <i>With</i>	64
9. Pętle i inne sposoby odwoływania się do komórek	66
9.1. Przykłady zastosowania właściwości <i>Cells</i>	66
9.2. Operacje na kolumnach i wierszach	67
9.3. Pętla skończona <i>For... To</i>)	68
9.4. Pętle nieskończone	71
10. Zaawansowane możliwości obiektu <i>Range</i>	76
10.1. Obiekty globalne typu <i>Range</i>	76
10.2. Przeglądarka obiektów	79
10.3. Komentarze – umieszczanie w kodzie informacji dla użytkownika	82
11. Procedury zdarzeń	83
11.1. Uruchamianie makra przy otwieraniu skoroszytu	83
11.2. Uruchamianie makr przy zamykaniu skoroszytu	85
11.3. Uruchamianie kodu przy zmianie wartości w komórkach	87
12. Formularze <i>UserForm</i> i podstawowe kontrolki	90
12.1. Tworzenie formularzy do wprowadzania danych	90

Wstęp

Praca z Excelem potrafi być czasochłonna. Tymczasem nasz kolega z pracy, siedzący przy biurku obok, potrafi przygotować raport w ciągu 5 minut, podczas gdy nam to samo zadanie zajmuje ponad godzinę. Jak to możliwe? Odpowiedź jest prosta: napisał własne makro, które wykonało za niego całą pracę lub znaczną jej część.

Makra to kod napisany w języku Visual Basic for Application (VBA) wykonujący polecenia aplikacji, do której jest dołączony. Wiele z tych poleceń jest niedostępnych z poziomu okna Excela. Makra pozwalają na automatyzację pracy z dokumentami Excela i nie tylko. Mogą wykonywać tysiące poleceń za jednym razem i dlatego są niejako rozszerzeniem możliwości pracy z aplikacją, do której został dołączony edytor Visual Basic.

Jak w przypadku każdego kursu, w pierwszej kolejności należy poznać niezbędne podstawy, które pozwolą później swobodnie budować własne narzędzia w języku Visual Basic. Pierwsze rozdziały wyjaśniają więc podstawowe pojęcia, opisują najprostsze makra i podpowiadają, jak je uruchamiać. W kolejnych rozdziałach znajdziemy coraz bardziej zaawansowane zagadnienia, jak tworzenie pętli czy instrukcji warunkowych. Są w nich również bardzo złożone przykłady makr, które mogą przydać się w pracy. Zobaczymy, np. jak utworzyć formularz pomocny przy wypełnianiu zestawienia z fakturami. Zawarte w książce makra można wykorzystać w Excelu 2007, 2010 oraz 2013. Powinny również działać we wcześniejszych wersjach programu, choć mogą wymagać drobnych modyfikacji.



**Wszystkie pliki Excela z przykładami
omawianymi w książce można pobrać
ze strony:**

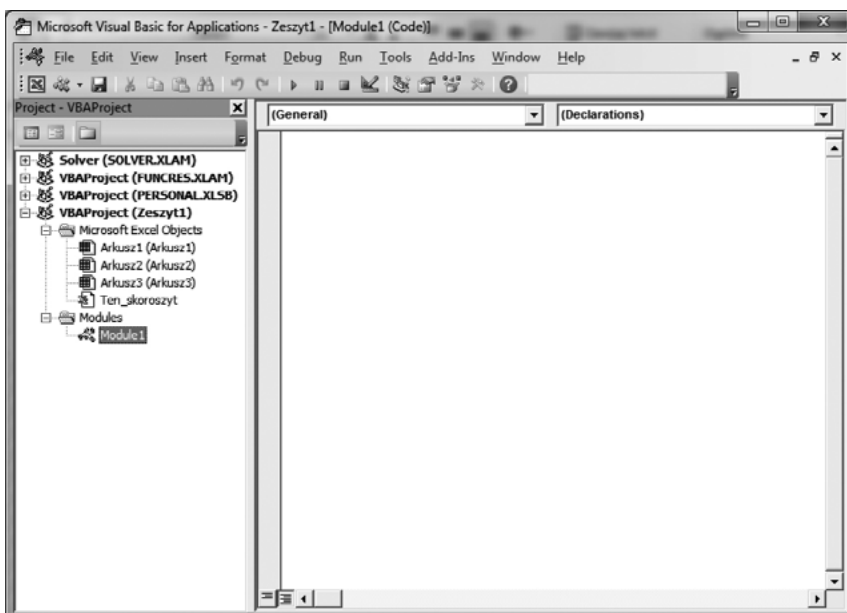
<http://online.wip.pl/download/exceltom12.zip>.

1. Informacje wstępne

Aby rozpocząć pracę z makrami, należy uruchomić edytor *Visual Basic*. We wszystkich programach pakietu Office można to zrobić, naciskając jednocześnie na klawiaturze lewy [Alt]+[F11]. Można również otworzyć kartę *Deweloper* i kliknąć *Visual Basic*. Jeśli nie widać karty *Deweloper*, przejdźmy do *Plik/Opcje/Dostosowywanie wstążki*. Na liście *Karty główne* zaznaczmy *Deweloper*.

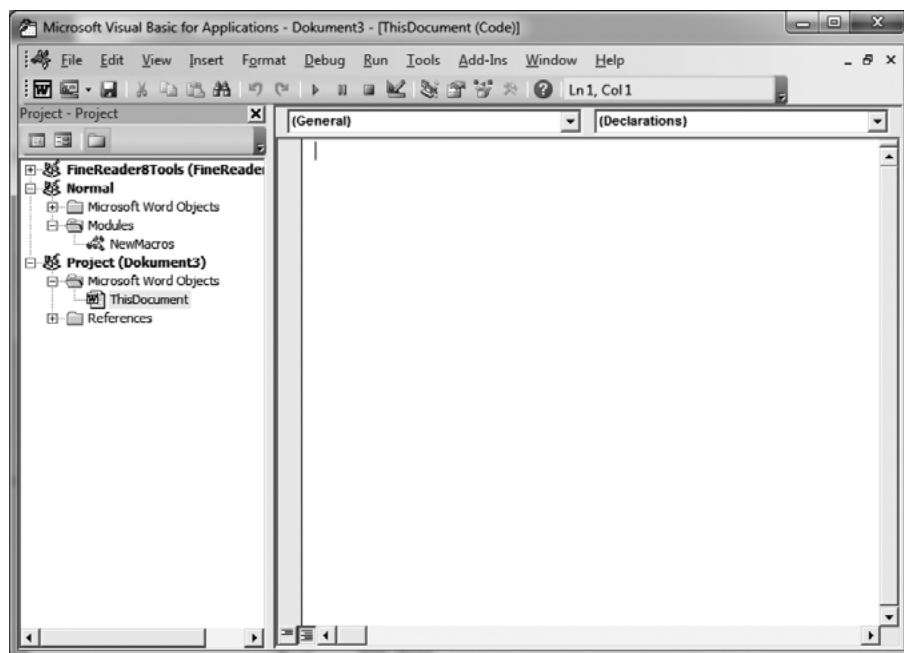
U góry znajduje się menu poleceń i pasek narzędzi *Standard*. Po lewej stronie u góry znajduje się okno przeglądarki projektu (*Project VBAProject*). W Excelu znajduje się tam jeden projekt o nazwie dokładnie takiej samej jak nazwa skoroszytu. Znajdują się w nim moduły wszystkich arkuszy zawartych w skoroszytu i dodatkowo moduł noszący nazwę *Ten_skoroszyt*. W niektórych komputerach okno przeglądarki projektu może zawierać też inne moduły czy projekty, w zależności od tego, czy korzystano już z makr lub instalowano dodatki.

Jeśli w edytorze nie widać przeglądarki projektu i okna właściwości (jak na rysunku 1.1), wciśnijmy najpierw kombinację klawiszy [Ctrl]+[R], a następnie [F4].



Rysunek 1.1. Okno edytora Visual Basic po otwarciu w Excelu

W programie Word (rysunek 1.2) w oknie przeglądarki projektów znajdują się dwa projekty: jeden o nazwie *Normal* i drugi o nazwie dokładnie takiej samej jak otwarty dokument. Ten drugi projekt zawiera moduł *ThisDocument*. W innych programach mających edytor Visual Basic jest jeszcze inaczej. O poszczególnych modułach w projektach dowiemy się więcej w kolejnych rozdziałach.



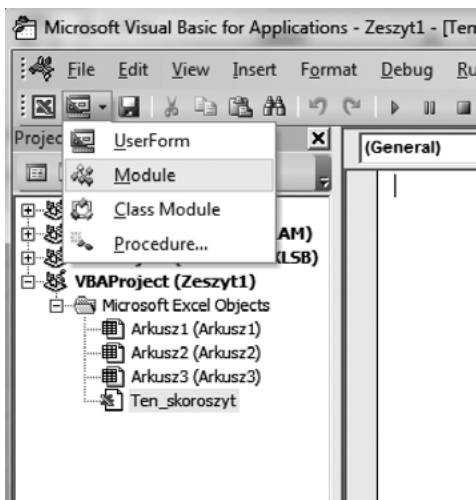
Rysunek 1.2. Okno edytora Visual Basic po otwarciu w Wordzie

Okno po lewej stronie poniżej przeglądarki projektu to okno właściwości. Więcej na temat właściwości dowiemy się w kolejnych rozdziałach. W tym pokażemy, jak utworzyć pierwsze, własne makro.

1.1. Przygotowanie własnej procedury

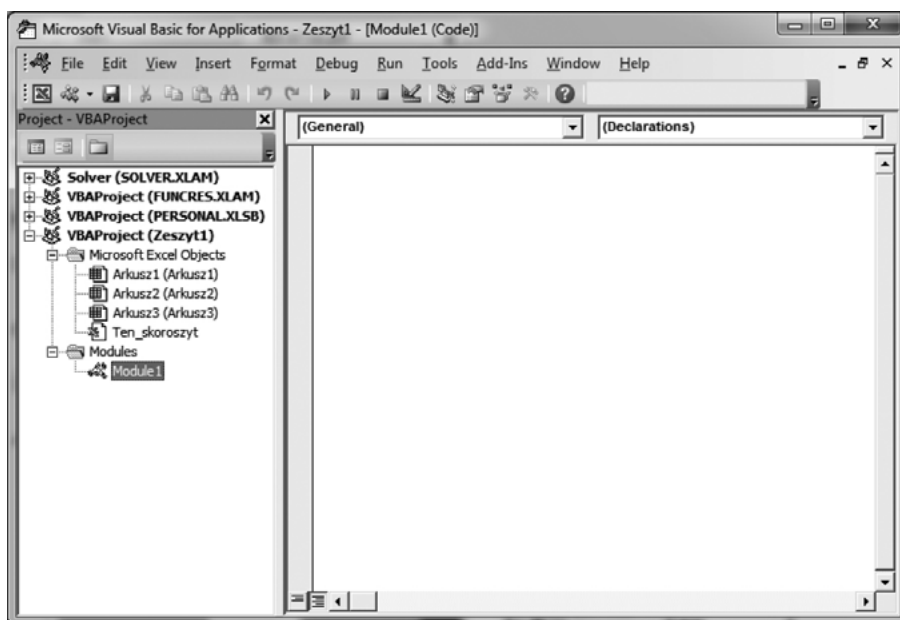
Makra z reguły są przechowywane w modułach standardowych. Aby wstawić taki moduł do projektu, należy w edytorze Visual Basic z menu *Insert* wybrać polecenie *Module* lub skorzystać z paska narzędzi *Standard*, tak jak to pokazano na rysunku 1.3.

Po wybraniu tego polecenia w oknie edytora Visual Basic pojawi się białe okno, a w oknie przeglądarki projektu zaznaczony będzie moduł standardowy o nazwie *Module1* (rysunek 1.4). Białe okno na środku to okno kodu, do którego należy wpisywać polecenia języka Visual Basic.



Rysunek 1.3. Wstawianie modułu standardowego

Makra zawarte są najczęściej wewnątrz procedury, która stanowi blok instrukcji języka Visual Basic. Każda procedura ma nazwę. Nazwa nie może zawierać spacji ani innych niedozwolonych znaków. Jeśli makro będzie wykorzystywane w aplikacjach obcojęzycznych, to nazwa nie powinna również zawierać polskich znaków.



Rysunek 1.4. Okno edytora Visual Basic po wstawieniu modułu standardowego