Witold Wrotek

VBA

dla Excela 2021 PL i 365 PL

234 praktyczne przykłady

Helion

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite / Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki
Grafika na okładce została wykorzystana za zgodą Shutterstock.com

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)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/vbe213
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-8898-7

Printed in Poland.

Spis treści

Rozdział 6.

Korzystanie z obiektów

Z tego rozdziału dowiesz się:

- Co to jest obiekt
- Co to jest właściwość
- Co to jest metoda

Obiekty

Obiektem jest element składowy aplikacji, np. komórka, zakres komórek, wykres umieszczony na arkuszu, arkusz, skoroszyt, arkusz programu Excel.

Excel zawiera obiekty skoroszytów, skoroszyty zawierają arkusze, arkusze zawierają zakresy, zakresy zawierają komórki. Powstaje więc struktura hierarchiczna.

Właściwości

Obiekty mają właściwości. Właściwość jest cechą obiektu. Właściwościami zakresu komórek (obiektu `Range`) są: zawartość komórki, styl czcionki itp.

Przy odwoływaniu się do właściwości danego obiektu używa się składni:

`Object``.`*`Property`*

gdzie:

- *`Object`* — nazwa obiektu,
- *`Property`* — nazwa właściwości.

Metody

Z obiektami są związane metody. Metoda jest operacją wykonaną na obiekcie — stanowi ją np. wpisanie tekstu do komórki czy zaznaczenie zakresu komórek.

Metoda jest działaniem, które można wykonać na obiekcie. Jest sposobem postępowania, który prowadzi do określonego rezultatu.

W celu posłużenia się metodą należy za nazwą obiektu wstawić kropkę, a następnie nazwę wybranej metody.

Zaznaczanie komórki

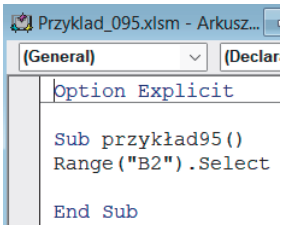
Do zaznaczania komórek można wykorzystać metodę `Select`.

Przykład 95.

Napisz program, którego zastosowanie pozwoli zaznaczyć komórkę *B2*.

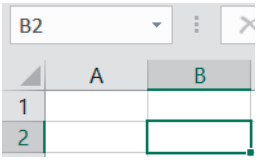
1. W arkuszu Excel zaznacz komórkę *A1*.
2. Wyświetl okno *Code*.
3. Wpisz w nim kod programu (rysunek 6.1). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej zapisano obiekt (`Range("B2")`). W ten sposób oznaczono pojedynczą komórkę *B2*. Następnie po kropce została zapisana metoda (`Select`), która umożliwia zaznaczenie komórki.

Rysunek 6.1. *Program, którego zastosowanie pozwala zaznaczyć komórkę B2*



4. Uruchom program.
5. W oknie `Macros` zaznacz `przyklad95`.
6. Wyświetl arkusz Excela (rysunek 6.2).

Rysunek 6.2. *Zaznaczenie zostało przeniesione do komórki B2*

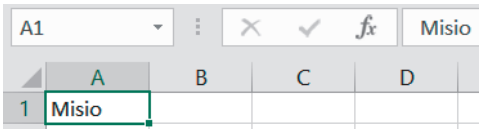


Przykład 96.

Napisz program, którego zastosowanie pozwoli wyczyścić komórkę *A1*.

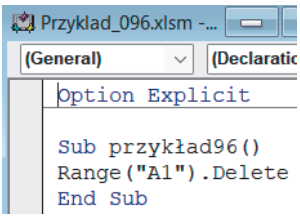
1. W komórce *A1* wpisz *Misio* (rysunek 6.3).

RYSUNEK 6.3. *Dane do testowania*



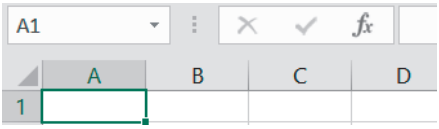
2. Wyświetl okno *Code*.
3. Wpisz w nim kod programu (rysunek 6.4). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej zapisano obiekt (`Range("A1")`). W ten sposób „pokazano" programowi, że jego działanie ma dotyczyć komórki *A1*. Następnie po kropce została zapisana metoda (`Delete`), która umożliwia usunięcie zawartości komórki.

RYSUNEK 6.4. *Program, którego zastosowanie pozwala wyczyścić komórkę B2*



4. Uruchom program.
5. W oknie `Macros` zaznacz `przyklad96`.
6. Wyświetl arkusz Excel (rysunek 6.5).

RYSUNEK 6.5. *Zawartość komórki A1 została bezpowrotnie utracona*



Elektroniczny sufler

Podczas pisania programu dużo problemów sprawiają:

- ♦ potrzeba dokładnej znajomości nazw obiektów i metod,
- ♦ literówki popełniane przy ich wpisywaniu.

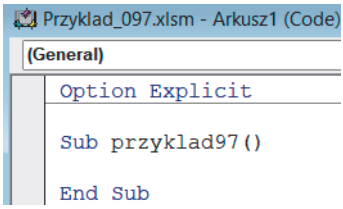
Na szczęście edytor VBA został wyposażony w system podpowiedzi. Jak z niego korzystać, dowiesz się z przykładu 97.

Przykład 97.

Napisz program, którego zastosowanie pozwoli zaznaczyć komórkę *B2*.

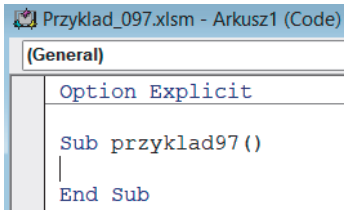
1. Wyświetl okno *Code*.
2. Utwórz szkielet programu. Linia kończąca procedurę pojawia się automatycznie po wprowadzeniu jej nagłówka (rysunek 6.6).

RYSUNEK 6.6. *Szkielet programu*



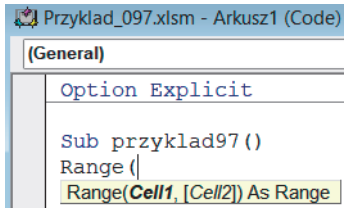
3. Umieść znak wstawiania między obiema liniami (rysunek 6.7).

RYSUNEK 6.7. *Znak wstawiania został umieszczony między liniami ograniczającymi procedurę*



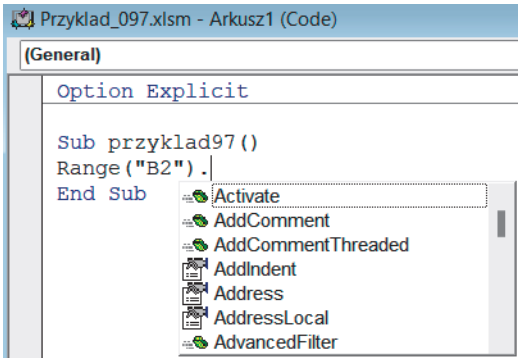
4. Podaj nazwę obiektu. W naszym przypadku będzie to zakres komórek. Wpisz `Range(`. W oknie edytora pojawiła się podpowiedź (rysunek 6.8).

RYSUNEK 6.8. *Edytor podpowiada, jak można zdefiniować zakres*

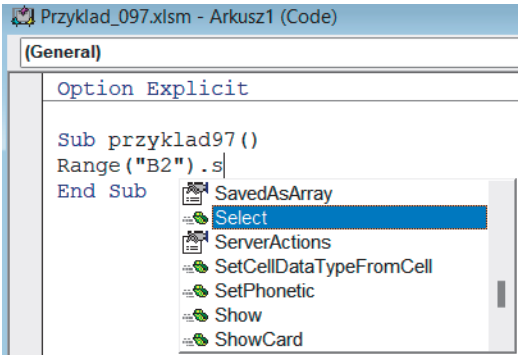


5. Zakresem będzie jedna komórka. Wpisz `"B2")`..
6. Po wpisaniu kropki w oknie edytora pojawiła się podpowiedź (rysunek 6.9).
7. Metody i właściwości są uporządkowane alfabetycznie. Zaznaczenie obiektu osiąga się poprzez zastosowanie metody `Select`. Nie musisz znać dokładnie jej pisowni. Wystarczy, że wiesz, na jaką literę zaczyna się nazwa. Wpisz literę `S`.
8. Została wyświetlona lista metod i właściwości o nazwach rozpoczynających się od litery *S*. Zaznacz metodę `Select` (rysunek 6.10).
9. Potwierdź wybór naciśnięciem klawisza *Enter*.

RYSUNEK 6.9. *Lista metod i właściwości*

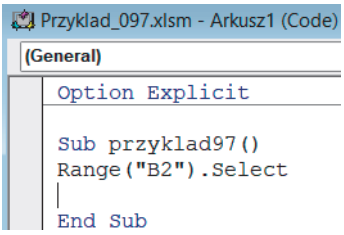


RYSUNEK 6.10. *Lista metod i właściwości o nazwach rozpoczynających się od litery S*



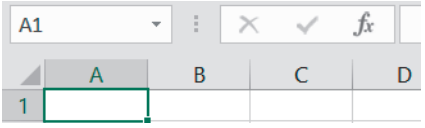
10. Do programu została wstawiona metoda `Select`. Program jest gotowy (rysunek 6.11).

RYSUNEK 6.11. *Program po dodaniu metody*



11. Teraz pora na sprawdzenie, czy program działa. Wyświetl okno arkusza Excel (rysunek 6.12). Domyślnie jest wybrana komórka *A1*.

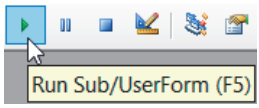
RYSUNEK 6.12. *Arkusz z wybraną komórką A1*



12. Wyświetl okno edytora.

13. Uruchom program kliknięciem ikony (rysunek 6.13).

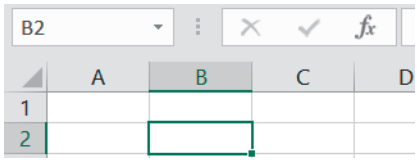
RYSUNEK 6.13.
Kliknięcie ikony spowoduje uruchomienie programu



14. Program został wykonany.

15. Wyświetl okno arkusza Excel (rysunek 6.14). Wybraną komórką jest *B2*.

RYSUNEK 6.14.
Arkusz z wybraną komórką B2



Usuwanie zawartości i formatowania

Usunięcie zawartości i formatowania z komórek wymaga zaznaczenia zakresu komórek i posłużenia się metodą `Clear`.

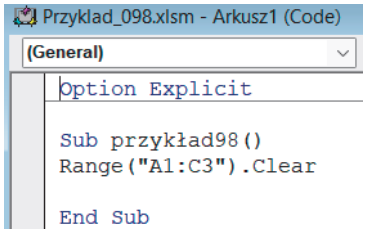
Przykład 98.

Napisz program, którego zastosowanie pozwoli czyścić i przywracać domyślne formatowanie komórek w zakresie od *A1* do *C3*.

1. Wyświetl okno *Code*.

2. Wpisz w nim kod programu (rysunek 6.15). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej znajduje się komenda czyszczenia zawartości i przywracania domyślnego formatowania komórek arkusza leżących w zakresie od *A1* do *C3*.

RYSUNEK 6.15.
Program, ktorego zastosowanie pozwala usunąć zawartość i formatowanie komórek w obszarze od A1 do C3



3. Wyświetl okno arkusza Excel. Domyślnie wszystkie komórki są puste. Wpisz dane do komórek (rysunek 6.16).

4. Wyświetl okno edytora.

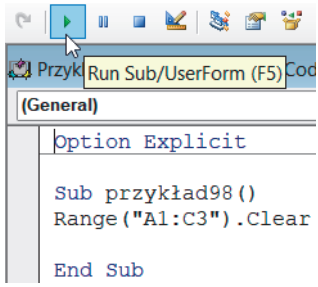
5. Uruchom program kliknięciem ikony (rysunek 6.17).

6. Program został wykonany.

RYSUNEK 6.16. *Arkusz z danymi*

	A	B	C	D
1	a	a	a	a
2	a	a	a	a
3	a	a	a	a
4	a	a	a	a

RYSUNEK 6.17. *Kliknięcie ikony spowoduje uruchomienie programu*



7. Wyświetl okno arkusza Excel (rysunek 6.18). Komórki z zakresu od *A1* do *C3* zostały wyczyszczone.

RYSUNEK 6.18. *Z komórek w zakresie od A1 do C3 zostały usunięte zawartość i formatowanie*

	A	B	C	D
1				a
2				a
3				a
4	a	a	a	a

Usuwanie zawartości

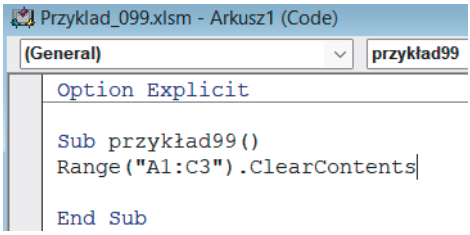
Usunięcie zawartości z komórek wymaga zaznaczenia zakresu komórek i posłużenia się metodą `ClearContents`.

Przykład 99.

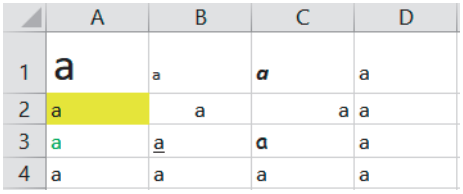
Napisz program, którego zastosowanie pozwoli usunąć zawartość komórek w zakresie od *A1* do *C3*.

1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.19). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej znajduje się komenda usuwania zawartości komórek arkusza leżących w zakresie od *A1* do *C3*.
3. Wyświetl okno arkusza Excel. Domyślnie wszystkie komórki są puste. Wpisz dane do komórek (rysunek 6.20).
4. Wyświetl okno edytora.

RYSUNEK 6.19. *Program, którego zastosowanie pozwala usunąć zawartość komórek w obszarze od A1 do C3*

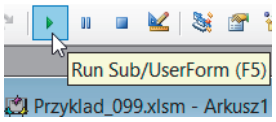


RYSUNEK 6.20. *Arkusz z danymi*



5. Uruchom program kliknięciem ikony (rysunek 6.21).

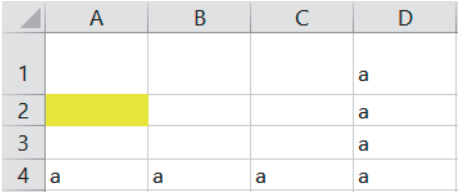
RYSUNEK 6.21. *Kliknięcie ikony spowoduje uruchomienie programu*



6. Program został wykonany.

7. Wyświetl okno arkusza Excel (rysunek 6.22). Z komórek z zakresu od *A1* do *C3* została usunięta zawartość.

RYSUNEK 6.22. *Z komórek z zakresu od A1 do C3 została usunięta zawartość, ale nie formatowanie*



Usuwanie formatowania

Zastąpienie dotychczasowego formatowania przez domyślne formatowanie komórek wymaga zaznaczenia zakresu komórek i posłużenia się metodą `ClearFormats`.

Przykład 100.

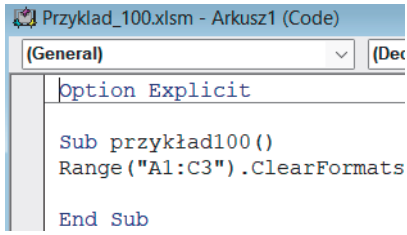
Napisz program, którego zastosowanie pozwoli przywrócić domyślne formatowanie komórek w zakresie od *A1* do *C3*.

1. Wyświetl okno *Code*.

2. Wpisz w nim kod programu (rysunek 6.23). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury.

W linii trzeciej znajduje się komenda usuwania formatowania komórek arkusza leżących w zakresie od *A1* do *C3*.

RYSUNEK 6.23. *Program, którego zastosowanie pozwoli usunąć formatowanie komórek w ograniczonym obszarze*



3. Wyświetl okno arkusza Excel. Domyślnie wszystkie komórki są puste. Wpisz dane do komórek (rysunek 6.24).

RYSUNEK 6.24. *Arkusz z danymi*

	A	B	C	D
1	a	a	a	a
2	a	a	a	a
3	a	a	a	a
4	a	a	a	a

4. Wyświetl okno edytora.
5. Uruchom program. Naciśnij klawisz *F5*.
6. Program został wykonany. Wyświetl okno arkusza Excel (rysunek 6.25). Z komórek z zakresu od *A1* do *C3* zostało usunięte formatowanie.

RYSUNEK 6.25. *Z komórek z zakresu od A1 do C3 zostało usunięte formatowanie, ale nie dane*

	A	B	C	D
1	a	a	a	a
2	a	a	a	a
3	a	a	a	a
4	a	a	a	a

Usuwanie wartości mniejszych od progowej

Do usunięcia zawartości komórek służy metoda `ClearContents`. Użycie jej tylko w stosunku do komórek o wartości mniejszej od progowej wymaga sprawdzania wartości komórek o zadanym zakresie.

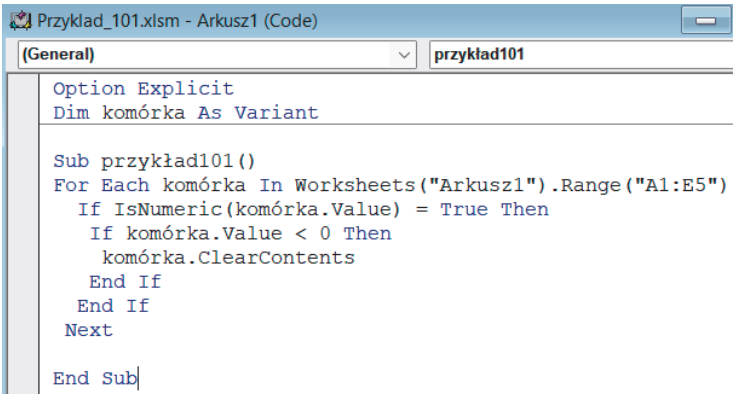
Przykład 101.

Napisz program, którego zastosowanie pozwoli przeszukiwać *Arkusz1* w zakresie od *A1* do *E5*. Zawartość komórek o wartościach mniejszych od 0 będzie usuwana.

1. Wyświetl okno *Code*.

2. Wpisz w nim kod programu (rysunek 6.26). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej została zadeklarowana zmienna `komórka`. Ponieważ nie wiadomo, jakie dane może zawierać arkusz, został zadeklarowany typ `Variant`. W trzeciej linii znajduje się nazwa procedury. Za pomocą pętli `For Each… Next` będzie przeszukiwany zakres komórek od *A1* do *E5* arkusza Excela o nazwie *Arkusz1*. Przy użyciu metody `ClearContents` zawartość komórek o wartościach mniejszych od `0` jest usuwana.

RYSUNEK 6.26. *Program, którego zastosowanie pozwala usunąć komórki o wartości mniejszej od 0 Arkusza1 w zakresie od A1 do E5*



3. Wyświetl okno arkusza Excel. W *Arkusz1* wpisz dane (rysunek 6.27).

RYSUNEK 6.27. *Dane wpisane do Arkusza1*

	A	B	C	D	E
1	-1	0	2	-33	
2	8	-1	0	4	
3	a	14.VII.1410	0,1	-0,02	
4	22		9	0.2	
5					

4. Wyświetl okno edytora.

5. Uruchom program. Naciśnij klawisz *F5*.

6. Program został wykonany. Wyświetl okno arkusza Excel (rysunek 6.28). Komórki z zakresu od *A1* do *E5* o wartościach mniejszych od zera zostały pozbawione wartości. Ich formatowanie pozostało niezmienione.

RYSUNEK 6.28. *Z komórek z zakresu od A1 do E5 zostały usunięte dane o wartościach mniejszych niż zero*

	A	B	C	D	E
1		0	2		
2	8		0	4	
3	a	14.VII.1410	0,1		
4	22		9	0.2	
5					

Właściwości

Właściwością jest cecha obiektu. Może nią być np. liczba wpisana w komórkę, kolor komórki itp.

Przypisanie wartości komórce

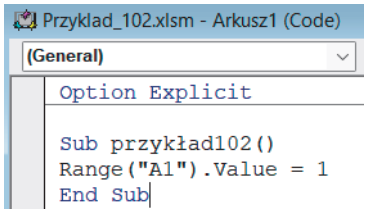
Właściwość, która jest przypisana komórce, zależy od wartości `Value`.

Przykład 102.

Napisz program, którego zastosowanie pozwoli wpisać liczbę 1 do komórki *A1*.

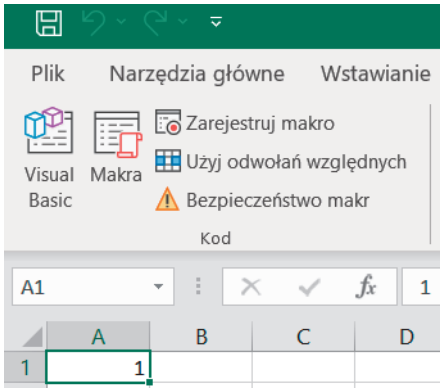
1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.29). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej zakresowi komórek jest przypisywana wartość. Jako zakres została podana jedna komórka, o adresie *A1*. Zostaje jej przypisana wartość 1.

Rysunek 6.29. *Do komórki A1 zostanie wpisana wartość 1*



3. Uruchom program. Naciśnij klawisz *F5*.
4. Program został wykonany. Do komórki *A1* została wstawiona liczba 1 (rysunek 6.30).

Rysunek 6.30. *Arkusz z liczbą wpisaną przez program. Operacji nie można cofnąć, ale po wydaniu polecenia zamknięcia arkusza program pyta o zapisanie wprowadzonych zmian*

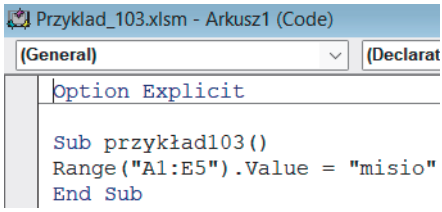


Przykład 103.

Napisz program, którego zastosowanie pozwoli wpisywać słowo `misio` do komórek w zakresie od *A1* do *E5*.

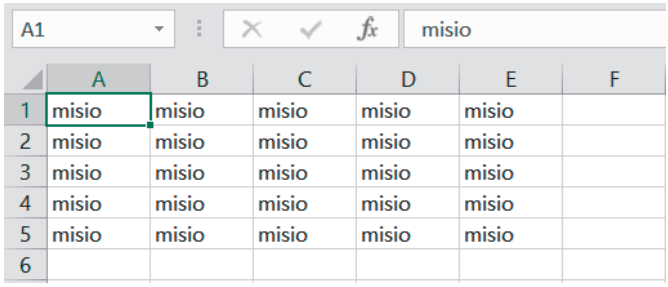
1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.31). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej komórkom należącym do zakresu od *A1* do *E5* jest przypisywana wartość. Wartością jest ciąg znaków — słowo `misio`.

Rysunek 6.31. *Komórki należące do zakresu od A1 do E5 zostaną wypełnione identycznym ciągiem znaków*



3. Uruchom program. Naciśnij klawisz *F5*.
4. Program został wykonany. Do komórek od *A1* do *E5* został wstawiony ciąg znaków `misio` (rysunek 6.32).

Rysunek 6.32. *Rezultatem wykonania programu jest błyskawiczne „rozmnożenie" słowa „misio"*



Zmian w arkuszu wprowadzonych przez uruchomiony program nie można cofnąć w prosty sposób, np. naciskając klawisze *Ctrl+Z*.

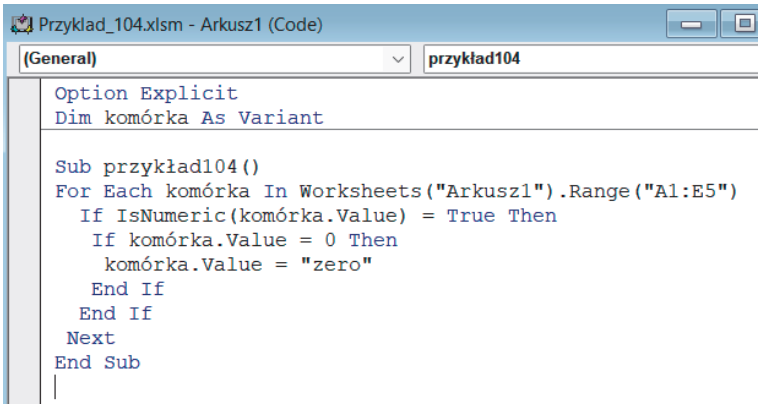
Przykład 104.

Napisz program, który będzie przeszukiwał *Arkusz1* w zakresie od *A1* do *E5*. W komórkach o wartościach równych zeru będzie wpisywane słowo `zero`.

1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.33). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii została zadeklarowana zmienna `komórka`. Ponieważ nie wiadomo, jakie dane może zawierać arkusz, został zadeklarowany typ `Variant`. W trzeciej linii znajduje się nazwa procedury.

Za pomocą pętli `For Each… Next` będzie przeszukiwany zakres komórek od *A1* do *E5* arkusza programu Excel o nazwie *Arkusz1*. Przy użyciu metody `Value` do komórek o wartościach równych `0` będzie wpisywane słowo `zero`.

RYSUNEK 6.33. *Program, za pomocą którego można zastąpić wartość liczbową jej słownym odpowiednikiem*



3. Wyświetl okno arkusza Excel. W *Arkusz1* wpisz dane (rysunek 6.34).

RYSUNEK 6.34. *Dane wpisane do Arkusza1*

	A	B	C	D	E
1	-1	0	2	-33	
2	8	-1	0	4	
3	a	14.VII.1410	0,1	-0,02	
4	22		9	0.2	
5					

4. Wyświetl okno edytora.
5. Uruchom program. Naciśnij klawisz *F5*.
6. Program został wykonany. Wyświetl okno arkusza Excel (rysunek 6.35). W komórki z zakresu od *A1*do *E5* o wartościach równych `0` został wpisany wyraz `zero`.

RYSUNEK 6.35. *Oto dowód na to, że wartość komórek pustych wynosi 0*

	A	B	C	D	E
1	-1	zero	2	-33	zero
2	8	-1	zero	4	zero
3	a	14.VII.1410	0,1	-0,02	zero
4	22	zero	9	0.2	zero
5	zero	zero	zero	zero	zero

Przykład 105. — do samodzielnego wykonania

Napisz program, za pomocą którego można będzie przeszukiwać *Arkusz1* w zakresie od *A1* do *E5*. W komórkach, w których znajduje się słowo `zero`, zostanie wpisana liczba `0`.

Przykład 106.

Napisz program, za pomocą którego będzie można przeszukiwać *Arkusz1* w zakresie od *A1* do *E5*. W komórkach, w których znajduje się liczba, zostanie wpisana jej wartość słownie.

1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.36). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii została zadeklarowana zmienna `komórka`. Ponieważ nie wiadomo, jakie dane może zawierać arkusz, został zadeklarowany typ `Variant`. W trzeciej linii znajduje się nazwa procedury. Za pomocą pętli `For Each… Next` będzie przeszukiwany zakres komórek od *A1* do *E5* arkusza Excel o nazwie *Arkusz1*. Przy użyciu metody `Value` do komórek o wartościach równych `0` będzie wpisywane słowo `zero`. Analogiczny warunek jest sprawdzany dla liczb o wartościach do 9 włącznie. Niespełnienie warunku powoduje, że będzie wykonywana kolejna linia programu. Spełnienie warunku powoduje, że do komórki będzie wpisywany ciąg znaków.
3. Wyświetl okno arkusza Excel. W *Arkusz1* wpisz dane (rysunek 6.37).
4. Wyświetl okno edytora.
5. Uruchom program. Naciśnij klawisz *F5*.
6. Program został wykonany. Wyświetl okno arkusza Excel (rysunek 6.38). W komórki z zakresu od *A1*do *E5* zostały wpisane wartości liczbowe wyrażone słownie.

Kopiowanie zawartości komórek

Kopiowanie zawartości komórek polega na przypisaniu komórkom należącym do jednego zakresu wartości komórek znajdujących się w innym zakresie. Operacja kopiowania jest możliwa dzięki nadaniu komórkom w obu zakresach identycznej właściwości `Value`.

Przykład 107.

Napisz program, za pomocą którego będzie można kopiować zawartość obszaru od *A1* do *A3* do obszaru od *B1* do *B3*.

1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.39). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii znajduje się nazwa procedury. W trzeciej linii obiektowi, którym jest zakres komórek od *B1* do *B3*, została przypisana zawartość komórek z zakresu od *A1* do *A3*.
3. Wyświetl arkusz Excela.
4. W komórki od *A1* do *A3* wpisz odpowiednio: `1`, `2`, `3` (rysunek 6.40).

RYSUNEK 6.36. *Program, za pomocą którego można zastąpić wartość liczbową z przedziału od 0 do 9 jej słownym odpowiednikiem*

Przyklad_106.xlsm - Arkusz1 (Code)

(General) | przykład106

```
Option Explicit
Dim komórka As Variant
Sub przykład106()
For Each komórka In Worksheets("Arkusz1").Range("A1:E5")
  If IsNumeric(komórka.Value) = True Then
   If komórka.Value = 0 Then
    komórka.Value = "zero"
   End If
   If komórka.Value = 1 Then
    komórka.Value = "jeden"
   End If
   If komórka.Value = 2 Then
    komórka.Value = "dwa"
   End If
   If komórka.Value = 3 Then
    komórka.Value = "trzy"
   End If
   If komórka.Value = 4 Then
    komórka.Value = "cztery"
   End If
   If komórka.Value = 5 Then
    komórka.Value = "pięć"
   End If
   If komórka.Value = 6 Then
    komórka.Value = "sześć"
   End If
   If komórka.Value = 7 Then
    komórka.Value = "siedem"
   End If
   If komórka.Value = 8 Then
    komórka.Value = "osiem"
   End If
   If komórka.Value = 9 Then
    komórka.Value = "dziewięć"
   End If
   End If
   Next
End Sub
```

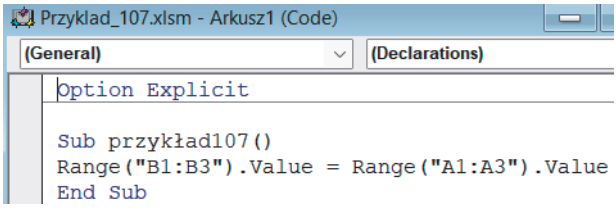
RYSUNEK 6.37. *Dane wpisane do Arkusza1*

	A	B	C	D	E
1	1				
2				3	
3					
4		2			
5					

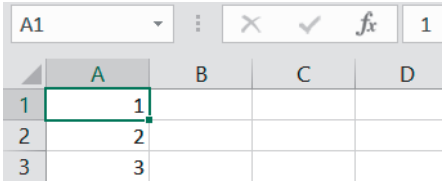
RYSUNEK 6.38. *Wartości liczbowe zostały wyrażone słownie*

	A	B	C	D	E
1	jeden	zero	zero	zero	zero
2	zero	zero	zero	trzy	zero
3	zero	zero	zero	zero	zero
4	zero	dwa	zero	zero	zero
5	zero	zero	zero	zero	zero

RYSUNEK 6.39. *Program, za pomocą którego będzie można zakresowi komórek od B1 do B3 przypisać wartość komórek od A1 do A3*

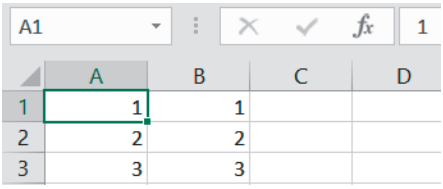


RYSUNEK 6.40. *Arkusz z zestawem danych*



5. Wyświetl okno edytora VBA.
6. Naciśnij klawisz *F5*.
7. Program został wykonany. Wyświetl arkusz Excela.
8. Do komórek od *B1* do *B3* została skopiowana zawartość komórek od *A1* do *A3* (rysunek 6.41).

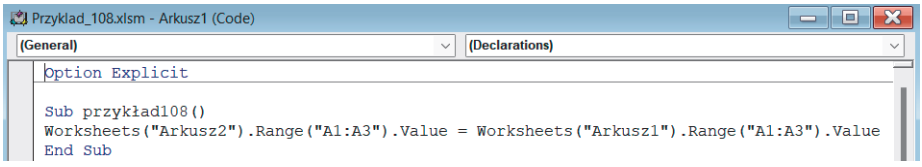
RYSUNEK 6.41. *W komórkach od B1 do B3 została wpisana zawartość komórek od A1 do A3*



Przykład 108.

Napisz program, za pomocą którego będzie można kopiować zawartość obszaru od *A1* do *A3* z arkusza 1. do obszaru od *A1* do *A3* arkusza 2.

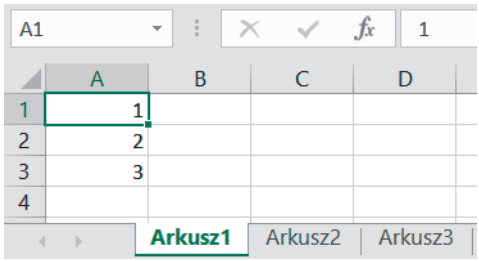
1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.42). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W linii drugiej znajduje się nazwa procedury. W linii trzeciej obiektowi, którym jest zakres komórek od *B1* do *B3*, została przypisana zawartość komórek z zakresu od *A1* do *A3*.



RYSUNEK 6.42. *Program, za pomocą którego będzie można pomiędzy arkuszami 1. i 2. skopiować wartość komórek od A1 do A3*

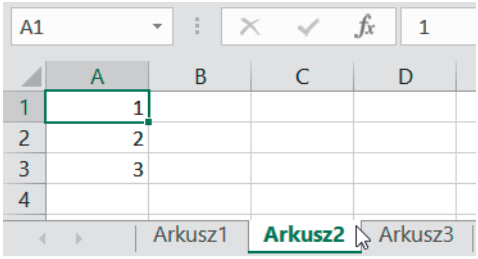
3. Wyświetl arkusz Excela.
4. W komórki od *A1* do *A3* wpisz odpowiednio: 1, 2, 3 (rysunek 6.43).

RYSUNEK 6.43. *Arkusz 1. z zestawem danych*

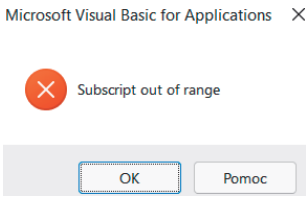


5. Wyświetl okno edytora VBA.
6. Naciśnij klawisz *F5*.
7. Program został wykonany. Wyświetl arkusz Excela i arkusz 2.
8. Do komórek od *A1* do *A3* arkusza 2. została skopiowana zawartość komórek od *A1* do *A3* arkusza 1. (rysunek 6.44).

RYSUNEK 6.44. *Arkusz 2. z zestawem danych skopiowanych z arkusza 1.*



Arkusz docelowy musi istnieć przed uruchomieniem programu. W przeciwnym razie wyświetlone zostanie okno z komunikatem błędu.



Nadawanie komórce koloru z użyciem nazwy koloru

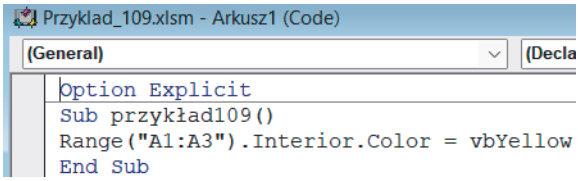
Aby nadać komórce kolor, wykorzystując nazwę koloru, należy posłużyć się właściwością `Interior.Color`.

Przykład 109.

Napisz program, za pomocą którego będzie można nadawać komórkom z obszaru od *A1* do *A3* kolor żółty.

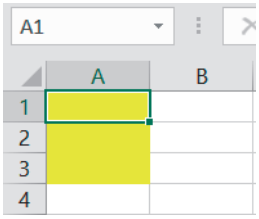
1. Uruchom arkusz kalkulacyjny Excel. Sprawdź, jaki kolor mają komórki. Domyślnie nie mają one żadnego wypełnienia.
2. Wyświetl okno *Code*.
3. Wpisz w nim kod programu (rysunek 6.45). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii znajduje się nazwa procedury. W trzeciej linii obiektowi, którym jest zakres komórek od *A1* do *A3*, zostaje przypisany kolor żółty.

Rysunek 6.45. *Komórki od A1 do A3 będą miały kolor żółty*



4. Uruchom program. Naciśnij klawisz *F5*.
5. Program został wykonany. Wyświetl arkusz kalkulacyjny Excel.
6. Kolor komórek od *A1* do *A3* został zmieniony (rysunek 6.46).

Rysunek 6.46. *Komórki, których kolor został zmieniony*



W tabeli 6.1 zebrano stałe, których przypisanie właściwości `Interior` powoduje wyświetlenie tła komórki w określonym kolorze.

Tabela 6.1. *Stałe odpowiadające najczęściej używanym kolorom*

Stała	Kolor
`vbBlack`	Czarny
`vbRed`	Czerwony
`vbGreen`	Zielony
`vbYellow`	Żółty
`vbBlue`	Niebieski

TABELA 6.1. *Stałe odpowiadające najczęściej używanym kolorom (ciąg dalszy)*

Stała	Kolor
`vbMagenta`	Fioletowy
`vbCyan`	Zielononiebieski
`vbWhite`	Biały

Nadawanie komórce koloru z użyciem kodu koloru

Szersze możliwości nadawania komórkom koloru stwarza posługiwanie się kodem koloru. Aby przypisać komórce kod koloru, należy posłużyć się właściwością `ColorIndex`.

Przykład 110.

Napisz program, którego zastosowanie pozwoli nadawać komórkom z obszaru od *A1* do *A56* kolory z palety barw dostępnej za pośrednictwem właściwości `ColorIndex`.

1. Uruchom arkusz kalkulacyjny Excel. Sprawdź, jaki kolor mają komórki. Domyślnie nie mają one żadnego wypełnienia.
2. Wyświetl okno *Code*.
3. Wpisz w nim kod programu:

```
Option Explicit
Sub przykład110()
Range("A1").Interior.ColorIndex = 1
Range("A2").Interior.ColorIndex = 2
Range("A3").Interior.ColorIndex = 3
Range("A4").Interior.ColorIndex = 4
Range("A5").Interior.ColorIndex = 5
Range("A6").Interior.ColorIndex = 6
Range("A7").Interior.ColorIndex = 7
Range("A8").Interior.ColorIndex = 8
Range("A9").Interior.ColorIndex = 9
Range("A10").Interior.ColorIndex = 10
Range("A11").Interior.ColorIndex = 11
Range("A12").Interior.ColorIndex = 12
Range("A13").Interior.ColorIndex = 13
Range("A14").Interior.ColorIndex = 14
Range("A15").Interior.ColorIndex = 15
Range("A16").Interior.ColorIndex = 16
Range("A17").Interior.ColorIndex = 17
Range("A18").Interior.ColorIndex = 18
Range("A19").Interior.ColorIndex = 19
Range("A20").Interior.ColorIndex = 20
Range("A21").Interior.ColorIndex = 21
Range("A22").Interior.ColorIndex = 22
Range("A23").Interior.ColorIndex = 23
Range("A24").Interior.ColorIndex = 24
Range("A25").Interior.ColorIndex = 25
Range("A26").Interior.ColorIndex = 26
Range("A27").Interior.ColorIndex = 27
```

```
Range("A28").Interior.ColorIndex = 28
Range("A29").Interior.ColorIndex = 29
Range("A30").Interior.ColorIndex = 30
Range("A31").Interior.ColorIndex = 31
Range("A32").Interior.ColorIndex = 32
Range("A33").Interior.ColorIndex = 33
Range("A34").Interior.ColorIndex = 34
Range("A35").Interior.ColorIndex = 35
Range("A36").Interior.ColorIndex = 36
Range("A37").Interior.ColorIndex = 37
Range("A38").Interior.ColorIndex = 38
Range("A39").Interior.ColorIndex = 39
Range("A40").Interior.ColorIndex = 40
Range("A41").Interior.ColorIndex = 41
Range("A42").Interior.ColorIndex = 42
Range("A43").Interior.ColorIndex = 43
Range("A44").Interior.ColorIndex = 44
Range("A45").Interior.ColorIndex = 45
Range("A46").Interior.ColorIndex = 46
Range("A47").Interior.ColorIndex = 47
Range("A48").Interior.ColorIndex = 48
Range("A49").Interior.ColorIndex = 49
Range("A50").Interior.ColorIndex = 50
Range("A51").Interior.ColorIndex = 51
Range("A52").Interior.ColorIndex = 52
Range("A53").Interior.ColorIndex = 53
Range("A54").Interior.ColorIndex = 54
Range("A55").Interior.ColorIndex = 55
Range("A56").Interior.ColorIndex = 56
End Sub
```

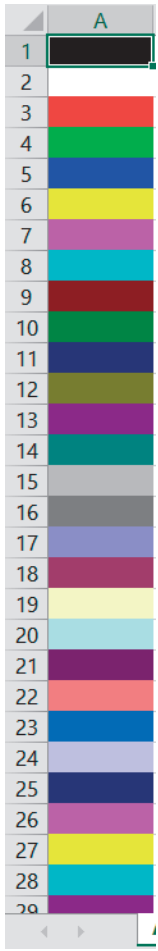
4. Uruchom program.
5. Wyświetl okno arkusza Excel.
6. W komórkach od *A1* do *A56* zostały wyświetlone próbki kolorów (rysunek 6.47).

Przykład 111.

Napisz program, którego zastosowanie pozwoli nadawać komórkom z obszaru od *A1* do *A56* kolory z palety barw dostępnej za pośrednictwem właściwości `ColorIndex`. Wykorzystaj instrukcję pętli.

1. Uruchom arkusz kalkulacyjny Excel. Sprawdź, jaki kolor mają komórki. Domyślnie nie mają one żadnego wypełnienia.
2. Wyświetl okno *Code*.
3. Wpisz w nim kod programu (rysunek 6.48). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii nazwa procedury `Dim wiersz As Integer` jest deklaracją zmiennej o nazwie `wiersz` typu `Integer`. Zmienna będzie przechowywała wartości licznika pętli. W linii `For wiersz=1 To 56` została zdefiniowana pętla. Przy pierwszym wykonaniu pętli wartość zmiennej `wiersz` równa się `1`. Ponieważ nie została określona wartość kroku pętli, jest przyjmowana wartość domyślna, czyli `1`.

RYSUNEK 6.47. *Numer wiersza odpowiada wartości parametru ColorIndex*



RYSUNEK 6.48. *Zamiast definiowania 56 operacji przypisania wartości, jak w poprzednim przykładzie, można posłużyć się instrukcją pętli*

Przyklad_111.xlsm - Arkusz1 (Code)

(General) (Declarations)

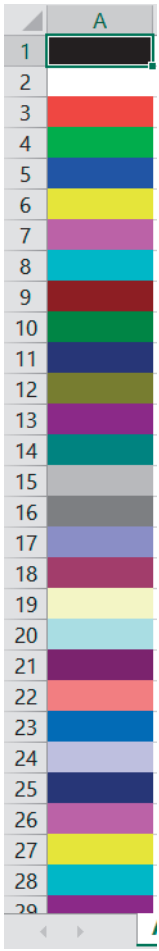
```
Option Explicit
Dim i As Integer
Sub przykład111()
 Dim wiersz As Integer
 For wiersz = 1 To 56
 Cells(wiersz, 1).Interior.ColorIndex = wiersz
 Next wiersz
End Sub
```

Wartość zmiennej `wiersz` jest zwiększana o wartość 1 przy każdym wykonaniu pętli.Pętla jest wykonywana ostatni raz, gdy wartość tej zmiennej osiągnie 56. Zmienna `wiersz` przybiera wartości numeryczne od 1 do 56 i pełni funkcję licznika pętli. Po podstawieniu do linii `Cells(wiersz,1).Interior.ColorIndex=wiersz` określa adres komórki i przypisany jej kod koloru. W przykładzie stały numer ma kolumna. Numer wiersza zaś jest określany przez zmienną `wiersz` i przy każdym wykonaniu

pętli jest zwiększany o jeden. Instrukcja `Next wiersz` powoduje zapętlenie. W instrukcji `Next` nazwa licznika (`wiersz`) nie jest wymagana. Dzięki podaniu nazwy licznika w instrukcji kończącej pętlę staje się ona bardziej czytelna.

4. Uruchom program.
5. Wyświetl okno arkusza Excel.
6. W komórkach od *A1* do *A56* zostały wyświetlone próbki kolorów (rysunek 6.49).

RYSUNEK 6.49. *Rezultat identyczny jak na rysunku 6.47 osiągnięto znacznie mniejszym nakładem pracy*



Nadawanie koloru zawartości komórki

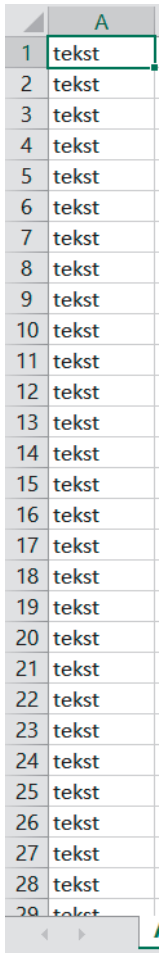
Kolorowanie tła komórek nie wyczerpuje wszystkich możliwości wyróżniania kolorem obszarów arkusza. Ciekawy efekt można uzyskać, zmieniając kolor czcionki. Aby przypisać czcionce kod koloru, należy posłużyć się właściwością `Font`.

Przykład 112.

Napisz program, którego zastosowanie pozwoli nadawać czcionkom z obszaru od *A1* do *A56* kolory z palety barw dostępnej za pośrednictwem właściwości `ColorIndex`. Wykorzystaj instrukcję pętli.

1. Uruchom arkusz kalkulacyjny Excel. Wpisz tekst do komórek od *A1* do *A56* (rysunek 6.50).

Rysunek 6.50. *Domyślnie tekst ma kolor czarny*

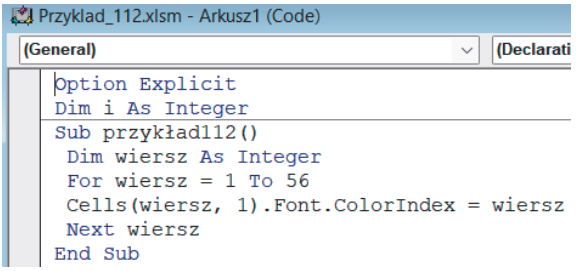


2. Wyświetl okno *Code*.

3. Wpisz w nim kod programu (rysunek 6.51). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii nazwa procedury `Dim wiersz As Integer` jest deklaracją zmiennej o nazwie `wiersz` typu `Integer`. Zmienna będzie przechowywała wartości licznika pętli. W linii `For wiersz=1 To 56` została zdefiniowana pętla. Przy pierwszym wykonaniu pętli wartość zmiennej `wiersz` równa się 1. Ponieważ nie została określona wartość kroku pętli, jest przyjmowana

wartość domyślna, czyli 1. Wartość zmiennej `wiersz` jest zwiększana o wartość 1 przy każdym wykonaniu pętli. Pętla jest wykonywana ostatni raz, gdy wartość tej zmiennej osiągnie 56. Zmienna `wiersz` przybiera wartości numeryczne od 1 do 56 i pełni funkcję licznika pętli. Po podstawieniu do linii `Cells(wiersz,1)` ↪`.Font.ColorIndex=wiersz` określa adres komórki i przypisany jej kod koloru. W przykładzie stały numer ma kolumna. Numer wiersza zaś jest określany przez zmienną `wiersz` i przy każdym wykonaniu pętli zwiększany o jeden. Instrukcja `Next wiersz` powoduje zapętlenie. W instrukcji `Next` nazwa licznika (`wiersz`) nie jest wymagana. Dzięki podaniu nazwy licznika w instrukcji kończącej pętlę staje się ona bardziej czytelna.

RYSUNEK 6.51. *Program, którego zastosowanie pozwoli zamienić napisy czarne w wielobarwne*



4. Uruchom program.
5. Wyświetl okno arkusza Excel.
6. W komórkach od *A1* do *A56* zostały wyświetlone próbki kolorów (rysunek 6.52).

Przesuwanie aktywnej komórki

W arkuszu kalkulacyjnym musi zostać wybrana przynajmniej jedna komórka. W niej będą się np. pojawiały znaki wpisywane z klawiatury. Jest ona wyróżniona grubszą obwódką.

Jest możliwe pozostawienie zaznaczenia jednej komórki i uczynienie aktywną innej. Jest to wykonalne dzięki właściwości `Offset`. Ma ona następującą składnię:

`Offset(`*x*`,`*y*`)`

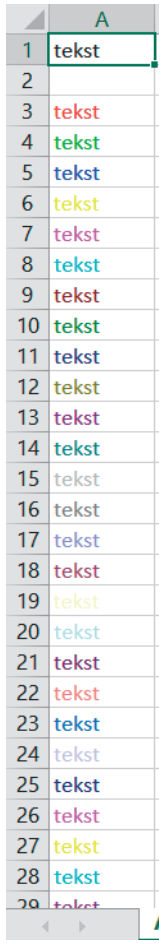
gdzie:

- *x* — przesunięcie w poziomie,
- *y* — przesunięcie w pionie.

Wartości ujemne pozwalają odpowiednio na przesunięcie w lewo lub w górę. Wartości dodatnie pozwalają odpowiednio na przesunięcie w prawo lub w dół.

Oczywiście istnieje również możliwość sterowania z poziomu programu VBA zaznaczeniem komórki. Umożliwia to właściwość obiektu `Application` — `ActiveCell`.

Rysunek 6.52. *Fragment arkusza z rysunku 6.50 po zmianie kolorów czcionki*

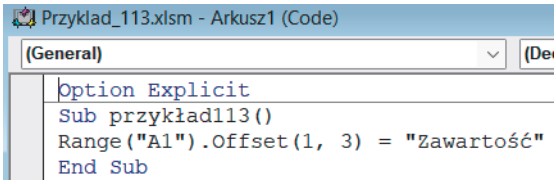


Przykład 113.

Napisz program, którego zastosowanie pozwoli przesuwać aktywną komórkę o jedną komórkę w dół i trzy komórki w prawo, a następnie wstawić do niej tekst `Zawartość`.

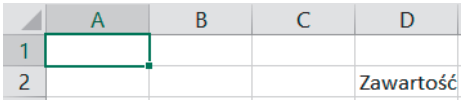
1. Wyświetl okno *Code* (rysunek 4.3).
2. Wpisz w nim kod programu (rysunek 6.53). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii jest widoczna nazwa procedury. W trzeciej linii znajduje się instrukcja zmiany aktywnej komórki. Punktem odniesienia jest komórka *A1*. Właściwość `Offset` pozwala na przesunięcie zaznaczenia o jedną komórkę w dół i trzy komórki w prawo. Ciąg znaków widoczny po znaku równości zostanie wstawiony do zaznaczonej komórki.
3. Uruchom program.
4. Wyświetl okno arkusza Excel.

RYSUNEK 6.53. *Program, którego zastosowanie pozwoli na wstawienie napisu do komórki, która nie jest zaznaczona*



5. W komórce *D2* pojawił się napis (rysunek 6.54).

RYSUNEK 6.54. *Jest zaznaczona komórka A1. Treść została wpisana do komórki odległej o zadane przesunięcie*



Przykład 114. — do samodzielnego wykonania

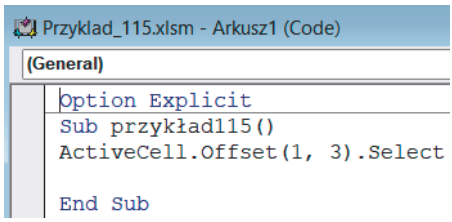
Napisz program, którego zastosowanie pozwoli przesuwać aktywną komórkę o trzy komórki w dół i jedną w prawo, a następnie wstawić do niej tekst `Zawartość`.

Przykład 115.

Napisz program, którego zastosowanie pozwoli przesuwać zaznaczenie komórki o jedną komórkę w dół i trzy komórki w prawo.

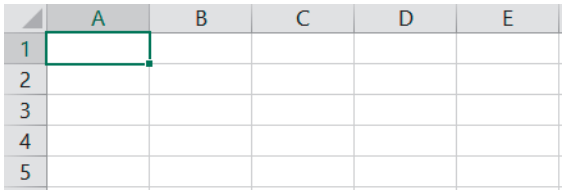
1. Wyświetl okno *Code*.
2. Wpisz w nim kod programu (rysunek 6.55). Pierwsza linia zawiera wymuszenie deklaracji wszystkich zmiennych. W drugiej linii jest widoczna nazwa procedury. W trzeciej linii znajduje się instrukcja wyboru zaznaczenia komórki. Punktem odniesienia jest bieżąca komórka. Właściwość `Offset(1, 3)` powoduje przesunięcie zaznaczenia względem aktualnego o jedną komórkę w dół i trzy komórki w prawo. Metoda `Offset` pozwala na wybranie komórki i oznaczenie jej pogrubioną ramką.

RYSUNEK 6.55. *Program, którego zastosowanie pozwala przesuwać zaznaczenie komórki o jedną komórkę w dół i trzy komórki w prawo*

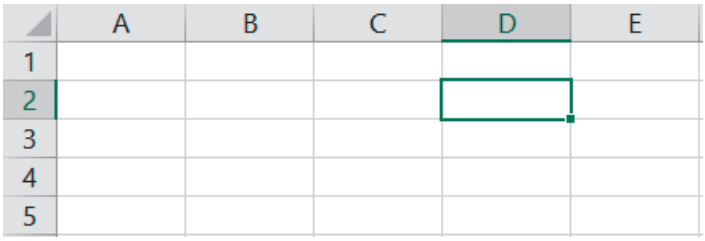


3. Wyświetl arkusz kalkulacyjny Excel. Sprawdź, czy komórka *A1* jest zaznaczona. Jeżeli nie — zaznacz ją (rysunek 6.56).
4. Uruchom program. Naciśnij klawisz *F5*.
5. Wyświetl okno arkusza Excel (rysunek 6.57).

Rysunek 6.56. *Komórka zaznaczona domyślnie*



Rysunek 6.57. *Jest zaznaczona komórka D2*



Przykład 116. — do samodzielnego wykonania

Napisz program, który będzie „pytał" o współrzędne zaznaczonej komórki, a następnie „przesuwał" zaznaczanie zgodnie z wprowadzonymi danymi.

Podsumowanie

- Obiektem jest element składowy aplikacji, np. komórka, zakres komórek, wykres umieszczony na arkuszu, arkusz, skoroszyt.
- Obiekty mają właściwości. Właściwość jest cechą obiektu. Właściwościami zakresu komórek są: zawartość komórki, styl czcionki itp.
- Metoda jest operacją wykonaną na obiekcie, umożliwia np. wpisanie tekstu do komórki lub zaznaczenie zakresu komórek.

Skorowidz

E

F

G

H

S

Ś

T

U

V

W

X

Y

Z

Ź

Excel z VBA — nowy poziom wydajności

- **Odkryj język VBA i jego zastosowania**
- **Poznaj sztukę tworzenia makropoleceń**
- **Zautomatyzuj swoją pracę w Excelu**

Masz dość wykonywania ciągle tych samych zadań? Musisz pilnie opracować kolejny raport dla szefa? Od wpatrywania się w tabele w Excelu Twoje oczy robią się kwadratowe? A może masz już dość bycia zwykłym użytkownikiem aplikacji i chcesz czegoś więcej? Świetnie! Wsparcie nadchodzi!

Excel to niezwykle popularny i uniwersalny program umożliwiający przeprowadzanie obliczeń naukowych, statystycznych, finansowych, analizę danych, wizualizację wyników i tworzenie rozbudowanych raportów. Ciekawie robi się jednak dopiero wtedy, gdy zachodzi potrzeba automatyzacji działań i prowadzenia interakcji z użytkownikami arkusza lub innymi aplikacjami pakietu Microsoft Office. Ciekawie, a może raczej trudno? Spokojnie, wszystko masz pod ręką, wystarczy skorzystać z tego, co oferuje język Visual Basic for Applications.

Jeśli chcesz wejść na wyższy poziom i dowiedzieć się, jak upraszczać i przyspieszać pracę z Excelem, sięgnij po tę książkę! Dzięki niej szybko i łatwo poznasz możliwości makr, metody ich tworzenia, konstrukcje języka VBA i różne sposoby ich zastosowania. Wiedza nie jest tu oderwana od rzeczywistości, lecz przekazywana w odniesieniu do praktycznych przykładów z życia. A to oznacza, że nauczysz się nie tylko skutecznie programować, lecz również rozwiązywać prawdziwe problemy, z którymi na co dzień mierzy się wielu użytkowników Excela. Ruszaj w drogę, przygoda czeka!

- **Makropolecenia i język VBA w Excelu**
- **Korzystanie z edytora języka VBA**
- **Komunikacja z użytkownikiem aplikacji**
- **Zastosowanie zmiennych i obiektów arkusza**
- **Używanie instrukcji warunkowych i podprogramów**
- **Obsługa zdarzeń i korzystanie z zakresów arkusza**
- **Zastosowanie funkcji arkuszowych i elementów sterujących**

Automatyzacja pracy w Excelu? Tylko z VBA!

ISBN 978-83-283-8898-7

Cena: 69,00 zł