

# Funkcje wyszukiwania i adresu – PODAJ.POZYCJĘ

## Mariusz Jankowski

autor strony internetowej poświęconej Excelowi i programowaniu w VBA;

## Bogdan Gilarski

właściciel firmy szkoleniowej Perfect And Practical;  
Pytania: czytelnicy.controlling@infor.pl

W tym opracowaniu kontynuujemy problematykę funkcji wyszukiwania i adresu. W numerze 2/2010 miesięcznika opisaliśmy zastosowanie WYSZUKAJ.PIONOWO. Tym razem napiszemy o jednej z najszybszych, a także najbardziej użytecznych i wszechstronnych funkcji arkuszowych – PODAJ.POZYCJĘ.

Przeznaczenie funkcji PODAJ.POZYCJĘ zawarte jest niejako w jej nazwie. Jej głównym zadaniem jest zlokalizowanie w wyznaczonym obszarze poszukiwanej wartości lub – mówiąc prościej, określenie, którą pozycję na liście zajmuje szukana wartość. Listą w tym przypadku jest zawsze jednowymiarowa tablica (pionowa lub pozioma). Algorytmy przeszukiwania nie rozróżniają wielkości liter, pomijają puste komórki, a także umożliwiają korzystanie ze znaków specjalnych przy definiowaniu kryteriów dla szukanej wartości.

Funkcja ta działa na łączach międzyskoroszytowych (osobnych plikach Excela). Po określeniu takiego łącza źródło danych (skoroszyt zawierający tabelę, z której pobieramy informacje) może zostać zamknięte, czyli *de facto* możemy wyciągać dane np. z cennika umieszczonego w innym, aktualnie zamkniętym skoroszytcie.

Funkcja w wyniku zwraca zawsze liczbę naturalną (mniejszą lub równą od pozycji ostatniej wartości na liście) lub błąd (w sytuacji gdy nie można określić pozycji dla szukanej wartości).

W przeciwieństwie do WYSZUKAJ.PIONOWO funkcja PODAJ.POZYCJĘ zwraca pozycję/miejsce wartości na liście, a nie samą wartość, co sprawia, że jest ona często zagnieżdżana w innych funkcjach wyszukiwania i adresu, takich jak: INDEKS, ADR.POŚR, PRZESUNIECIE oraz funkcji z grupy WYSZUKAJ... Innymi słowy, można powiedzieć, że funkcja PODAJ.POZYCJĘ służy przeważnie do dostarczania parametrów (argumentów) dla innych funkcji. Omówienie aspektów współpracy funkcji PO-

DAJ.POZYCJĘ z poszczególnymi, wymienionymi powyżej funkcjami będzie przedmiotem kolejnych, odrębnych artykułów.

## Widok 1

### Uwarunkowania funkcji PODAJ.POZYCJĘ()

	A	B	C	D	E	F	G
1	Uwarunkowania funkcji PODAJ.POZYCJĘ()						
2							
3	styczeń	luty	marzec	kwiecień	maj		
4							
5	styczeń		3 =PODAJ.POZYCJĘ("marzec";\$A\$3:\$E\$3)				
6	luty		3 =PODAJ.POZYCJĘ("marzec";\$A\$5:\$A\$9)				
7	marzec		5 =PODAJ.POZYCJĘ("MAJ";\$A\$3:\$E\$3:0)				
8	kwiecień		3 =PODAJ.POZYCJĘ("marzec";{"styczeń";"luty";"Ma"})				
9	maj		1 =PODAJ.POZYCJĘ("eń";\$A\$5:\$A\$9:0)				
10							

Jak widać na **widoku 1**, funkcja:

- działa prawidłowo zarówno dla tabeli pionowej, jak i poziomej (formuła 1 i 2),
- nie rozróżnia wielkości liter zarówno dla szukanej wartości, jak i dla poszczególnych elementów tablicy (formuła 3 i 4),
- umożliwia korzystanie ze znaków specjalnych (formuła 4 i 5).

## Budowa funkcji

Budowa funkcji jest stosunkowo prosta i wymaga podania następujących trzech argumentów:

1. *Szukana wartość* to po prostu wartość, którą chcemy odszukać wewnątrz jednowymiarowej tabeli/tablicy. To, czego szukamy, przeważnie jest tekstem lub liczbą, ale może to być także wartość logiczna typu PRAWDA, FAŁSZ.

2. *Przeszukiwana\_tab* to tablica (wpisana ręcznie) lub tabela (ciągły zakres komórek).

3. *Typ\_porównania* (domyślnie 1) może przybierać wartości 0, -1 lub 1.

Wartość 0 oznacza, że funkcja stosuje algorytm dokładnego przypisania, czyli zwraca pozycję na liście, wartości określonej w pierwszym argumencie (*Szukana wartość*). Dane w tablicy mogą być uporządkowane w dowolny sposób.

Wartość 1 oznacza, że dane muszą być posortowane rosnąco (od najmniejszej do największej). Funkcja znajduje pierwszą większą wartość od szukanej i w wyniku zwraca pozycję wartości umieszczonej powyżej (tablica jednokolumnowa) lub po lewej (tablica jednowierszowa).

Wartość -1 oznacza, że dane muszą być posortowane malejąco (od największej do najmniejszej, tak jak to ma miejsce w tabeli powyżej). Funkcja znajduje pierwszą mniejszą wartość od szukanej i w wyniku zwraca pozycję wartości umieszczonej powyżej (tablica jednokolumnowa) lub po lewej (tablica jednowierszowa).

### Algorytm „dokładne dopasowanie”

Algorytm dokładnego dopasowania (ostatni argument ustawiony na 0) w praktyce jest wykorzystywany najczęściej. Jego działanie polega na wyszukaniu w tablicy/tabeli danych dokładnie takiej samej wartości, jaka jest określona w pierwszym argumencie funkcji (jak wspomnieliśmy, wielkość liter nie jest rozróżniana). Jeżeli szukana wartość znajduje się na jednowymiarowej tablicy, wówczas zwracana jest jej pozycja; w przeciwnym razie funkcja generuje błąd #N/D!

#### Widok 2

##### Algorytm „dokładne dopasowanie”

	A	B	C	D	E	F
1	Algorytm „dokładne dopasowanie”					
2						
3	0	10	20	30	40	
4						
5	styczeń		3 =PODAJ.POZYCJĘ(20;\$A\$3:\$E\$3;0)			
6	luty	#N/D!	=PODAJ.POZYCJĘ("4*";\$A\$3:\$E\$3;0)			
7	marzec		2 =PODAJ.POZYCJĘ("luty";\$A\$5:\$A\$9;0)			
8	kwiecień	#N/D!	=PODAJ.POZYCJĘ("czerwiec";\$A\$5:\$A\$9;0)			
9	maj		3 =PODAJ.POZYCJĘ("?a*";\$A\$5:\$A\$9;0)			
10						

Jak pokazuje widok 2:

● algorytm zwraca prawidłowy wynik, gdy szukana wartość znajduje się na liście (formuła 1 i 3),

● algorytm generuje błąd, gdy szukana wartość nie znajduje się na liście (formuła 4),

● algorytm pozwala na stosowanie znaków specjalnych, ale tylko dla wyrażeń tekstowych (formuła 5), w przeciwnym razie generowany jest błąd #N/D! (formuła 2).

### Algorytm „pierwsze większe”

Trzeci argument funkcji PODAJ.POZYCJĘ domyślnie przyjmuje wartość 1. Oznacza to, że aby funkcja zwracała prawidłowe wyniki, dane w tabeli muszą zostać wcześniej posortowane rosnąco. Algorytm znajduje pierwszą wartość większą od szukanej i przyporządkowuje jej pozycję wartości powyżej (w przypadku tablicy pionowej) lub pozycję wartości po lewej (w przypadku tablicy poziomej).

Warto podkreślić, że algorytmu tego używamy niemal wyłącznie dla danych typu liczbowego – po prostu stosowanie porównań matematycznych (>, <) dla danych typu tekstowego jest bardzo rzadko spotykane, a często wręcz niepożądane (z reguły stosuje się w takich przypadkach algorytm dokładnego przypisania).

#### Widok 3

##### Algorytm „pierwsze większe”

	A	B	C	D	E	F
1	Algorytm „pierwsze większe”					
2						
3	0	10	20	30	40	
4						
5		#N/D!	=PODAJ.POZYCJĘ(-5;\$A\$3:\$E\$3;1)			
6			1 =PODAJ.POZYCJĘ(0;\$A\$3:\$E\$3;1)			
7			2 =PODAJ.POZYCJĘ(10;\$A\$3:\$E\$3;1)			
8			5 =PODAJ.POZYCJĘ(40;\$A\$3:\$E\$3;1)			
9			5 =PODAJ.POZYCJĘ(45;\$A\$3:\$E\$3;1)			
10						

Jak pokazuje widok 3:

● liczby w tablicy poziomej są posortowane rosnąco, wartość ostatniego argumentu ustawiona jest na 1 (jest to wartość domyślna, możemy zatem także pominąć ten argument),

● algorytm szuka pierwszej większej wartości, na prawo od szukanej – następnie w wyniku zwraca pozycję pierwszej wartości po lewej,

● algorytm zwraca błąd #N/D! w sytuacji, gdy szukana wartość jest mniejsza od pierwszej wartości na liście (formuła 1). Funkcja znajduje pierwszą wartość większą od szukanej (0>-5), ale nie jest w stanie pobrać pozycji wartości znajdującej się po lewej (bo wartość mniejsza od 0 nie istnieje w naszej tablicy).

### Algorytm „pierwsze mniejsze”

Analogicznie jak w przypadku algorytmu „pierwsze większe”, zakres zastosowania tego algorytmu to praktycznie analiza porównawcza dla danych liczbowych. W tym przypadku dane w tabeli danych muszą być posortowane malejąco, a wartość ostatniego argumentu (*Typ\_porównania*) ustawiona na -1. Algorytm znajduje pierwszą wartość mniejszą od szukanej i przyporządkowuje jej pozycję wartości powyżej (w przypadku tablicy pionowej) lub pozycję wartości po lewej (w przypadku tablicy poziomej).

#### Widok 4

Algorytm „pierwsze mniejsze”						
	A	B	C	D	E	F
1	Algorytm „pierwsze mniejsze”					
2						
3	40	30	20	10	0	
4						
5		#N/D!	=PODAJ.POZYCJĘ(45;\$A\$3:\$E\$3;-1)			
6		1	=PODAJ.POZYCJĘ(40;\$A\$3:\$E\$3;-1)			
7		4	=PODAJ.POZYCJĘ(10;\$A\$3:\$E\$3;-1)			
8		5	=PODAJ.POZYCJĘ(0;\$A\$3:\$E\$3;-1)			
9		5	=PODAJ.POZYCJĘ(-5;\$A\$3:\$E\$3;-1)			
10						

Na podstawie widoku 4 widzimy, że:

- liczby w tablicy poziomej są posortowane malejąco (od największej do najmniejszej), wartość ostatniego argumentu ustawiona jest na -1,
- algorytm szuka pierwszej mniejszej wartości, na prawo od szukanej – następnie w wyniku zwraca pozycję pierwszej wartości po lewej,
- algorytm zwraca błąd #N/D! w sytuacji, gdy szukana wartość jest większa od pierwszej wartości na liście (formuła 1). Funkcja znajduje pierwszą wartość mniejszą od szukanej (40<45), ale nie jest w stanie pobrać pozycji wartości znajdującej się po lewej (bo 40 jest pierwszą wartością na liście).

### Ostatnia niepusta komórka

Funkcję PODAJ.POZYCJĘ możemy także wykorzystać w niestandardowy sposób, aby np. pobrać pozycję ostatniej wartości na liście lub zlokalizować wiersz z ostatnią niepustą komórką w kolumnie. Jeżeli w pewnym zakresie danych występują puste komórki, możemy łatwo zlokalizować ostatnią niepustą. Wykorzystujemy w tym celu algorytm „pierwsze większe”, a jednocześnie jako szukaną podajemy ciąg tekstowy

(„zżz”), który na 100% jest wartością większą od wszystkich występujących w przeszukiwanym obszarze.

#### Widok 5

Ostatnia niepusta komórka					
	A	B	C	D	E
1	Inne przykłady zastosowania – ostatnia niepusta komórka				
2					
3	pierwszy wpis	4	=PODAJ.POZYCJĘ("zżz";\$A\$1:\$A\$5;1)		
4	drugi wpis	7	=PODAJ.POZYCJĘ("zżz";\$A\$1:\$A\$7;1)		
5		8	=PODAJ.POZYCJĘ("zżz";\$A\$1:\$A\$8;1)		
6	trzeci wpis				
7		-> spacja			
8		-> formuła zwracająca w wyniku pusty ciąg			
9					

Jak widać na widoku 5, w przypadku pierwszej formuły funkcja pomija ostatnią komórkę zakresu (ponieważ A5 jest pusta) i zwraca pozycję 4, czyli pozycję ostatniej niepustej komórki w zakresie. Należy jednak być bardzo uważnym, zarówno spacja jak i pusty ciąg zwracany przez formułę są przez funkcję PODAJ.POZYCJĘ traktowane jako komórki niepuste (odpowiednio formuła 2 i 3).

### Ostatnie wystąpienie z danych powtarzalnych

Wiemy już, że ustawiając wartość ostatniego argumentu *Typ\_porównania* na 0, otrzymujemy w wyniku pierwszą pozycję wystąpienia danej wartości (nawet gdy występuje ona na liście wiele razy). Algorytm „pierwsza większa” natomiast zwraca w wyniku pozycję pierwszej komórki mniejszej lub równej od szukanej. Fakt ten można czasami wykorzystać do wyszukania pozycji nie pierwszego, ale ostatniego wystąpienia danej wartości na liście. Warunkiem koniecznym w tym przypadku jest jednak, aby dane były posortowane.

Na widoku 6 widzimy powtarzające się wystąpienia niektórych dat (21-gru-2009 pojawia się na liście trzykrotnie), niektórych dat natomiast w tabeli nie ma (np. 22-gru-2009). W związku z tym pojawiają się dwa pytania:

- Jak określić ostatnie wystąpienie daty 21-gru-2009?
- Jak zabezpieczyć się przed błędem dla niewystępującej daty 22-gru-2009?

Po pierwsze należy bezwzględnie sprawdzić, czy szukana wartość znajduje się na liście (np. za pomocą formuły =JEŻELI(LICZ.JEŻELI(A3:

A9;C4)>0; PODAJ.POZYCJĘ(C4;A3:A9;1); "brak"). Jeżeli data nie może zostać zlokalizowana, funkcja generuje stosowny komunikat tekstowy; w przeciwnym razie zwraca pozycję 4, czyli pozycję nie pierwszego, lecz ostatniego wystąpienia wartości (21-gru-2009) na liście.

## Znaki specjalne w algorytmach wyszukiwania

Istnieje możliwość korzystania ze znaków specjalnych przy definiowaniu kryteriów dla szukanej wartości. Do najpopularniejszych można zaliczyć dwa takie symbole: „\*” zastępuje dowolny

ciąg tekstowy, „?” natomiast zastępuje jeden dowolny znak.

Już sam fakt zastosowania znaku specjalnego powoduje, że Excel w kryteriach porównania danych z szukanyymi (=, <, >) traktuje zmienne jako typ tekstowy. Może to mieć znaczenie w sytuacji, gdy będziemy chcieli zastosować znaki specjalne do pozycjonowania zmiennych typu liczbowego.

Jak wynika z **widoku 7**, pierwsza formuła szuka pozycji tekstu zaczynającego się na „3”, druga formuła szuka w tablicy tekstu zawierającego słowo „zakup”. Następne formuły korzystają

równocześnie z obydwu znaków specjalnych. Trzecia formuła zwraca pozycję tekstu, którego drugim znakiem jest „0”, czwarta zaś pozycję tekstu, który zaczyna się znakiem „0”, zawiera spację we właściwym miejscu, a kończy sylabą „je”. Piąta formuła zwraca błąd, ponieważ na liście nie istnieje tekst, którego pierwszy znak to „5”, a trzeci to „1”.

### Widok 6

#### Ostatnie wystąpienie z danych powtarzalnych

	A	B	C	D	E	F	G
1	Inne przykłady zastosowania – ostatnie wystąpienie z danych powtarzalnych						
2							
3	20-gru-2009						
4	21-gru-2009		21-gru-2009	4			
5	21-gru-2009		=JEŻELI(LICZ.JEŻELI(A3:A9;C4)>0; PODAJ.POZYCJĘ(C4;A3:A9;1); "brak")				
6	21-gru-2009						
7	23-gru-2009		22-gru-2009	brak			
8	23-gru-2009		=JEŻELI(LICZ.JEŻELI(A3:A9;C7)>0; PODAJ.POZYCJĘ(C7;A3:A9;1); "brak")				
9	24-gru-2009						
10							

### Widok 7

#### Znaki specjalne w algorytmach wyszukiwania

	A	B	C	D	E	F	G	H
1	Inne przykłady zastosowania – znaki specjalne w algorytmach wyszukiwania							
2								
3	010 Środki trwałe			3	=PODAJ.POZYCJĘ("3*";\$A\$3:\$A\$7;0)			
4	080 Inwestycje			5	=PODAJ.POZYCJĘ("**zakup*";\$A\$3:\$A\$7;0)			
5	310 Materiały			4	=PODAJ.POZYCJĘ("0*";\$A\$3:\$A\$7;0)			
6	400 Amortyzacja			2	=PODAJ.POZYCJĘ("0?? *je";\$A\$3:\$A\$7;0)			
7	503 Koszty zakupu		#N/D!		=PODAJ.POZYCJĘ("5?1*";\$A\$3:\$A\$7;0)			
8								

### Widok 8

#### Pozycjonowanie wyników działań na tablicach

	A	B	C	D	E	F	G	H	I
1	Inne przykłady zastosowania – pozycjonowanie wyników działań na tablicach								
2									
3	Sprzedaż Osoba			Sprzedaż	35				
4	10	Stefan		Osoba	Jan				
5	20	Jan							
6	20	Krzysztof		Pozycja	4				
7	50	Jan		=PODAJ.POZYCJĘ(1;(\$A\$4:\$A\$8>\$E\$3)*(\$B\$4:\$B\$8=\$E\$4);0)					
8	40	Stefan							
9									

## Pozycjonowanie wyników działań na tablicach

Funkcja PODAJ.POZYCJĘ() jest bardzo przydatna do szybkiego określenia pozycji, jaką zajmuje szukana wartość (spełniająca równocześnie kilka kryteriów, a nie jak to ma miejsce zazwyczaj, spełniająca tylko jeden warunek) na dynamicznej tablicy (utworzonej poprzez działanie formuły tablicowej). **Widok 8** ilustruje taką zależność.

W tabeli mamy zestawienie sprzedaży dokonanej przez pewne osoby. Zadanie polega na tym, aby pobrać pozycję pierw-



szej transakcji Jana (pierwszy warunek), dla sprzedaży przekraczającej kwotę 35 jednostek (drugi warunek).

Przykładowa formuła (zatwierdzamy Ctrl+Shift++Enter) rozwiązująca ten problem to: =PODAJ.POZYCJĘ(1;(\$A\$4:\$A\$8>\$E\$3)\*(\$B\$4:\$B\$8=\$E\$4);0)

gdzie właściwe zapytanie tablicowe o wyżej opisane warunki to formuła tablicowa:

=(\$A\$4:\$A\$8>\$E\$3)\*(\$B\$4:\$B\$8=\$E\$4);

dla danych E3 = 35 oraz E4 = „Jan” otrzymujemy tablicę wynikową: {0\0\0\1\0}, gdzie 1 i 0 to matematyczne wyniki działań na zmiennych typu logicznego (PRAWDA, FAŁSZ).

Jak łatwo sprawdzić, czwarty wiersz tabeli spełnia warunki zapytania.

Nałożenie na tę tablicę wynikową funkcji PODAJ.POZYCJĘ(), czyli:

=PODAJ.POZYCJĘ(1; {0\0\0\1\0}; 0)

określa nam dokładną pozycję jedynki w tabeli wynikowej i jednocześnie wskazuje numer jej wiersza.

## Wnioski

Przeznaczenie funkcji PODAJ.POZYCJĘ zawarte jest w jej nazwie. Służy ona do określenia pozycji (pozycjonowania), którą zajmuje szukana wartość na jednowymiarowej tablicy z danymi. Funkcja jest bardzo szybka i ma uniwersalne zastosowanie – bywa często wykorzystywana jako argument dla innych funkcji wyszukiwania i adresu. Nie rozróżnia ona wielkości liter, pozwala natomiast na stosowanie znaków specjalnych.

Oprócz standardowych właściwości funkcję PODAJ.POZYCJĘ można także wykorzystać do pobrania pozycji ostatniego wystąpienia danej liczby (daty) na liście bądź też określenia pozycji ostatniej niepustej komórki na liście. ■

### OD REDAKCJI

Wszystkie omawiane w tym opracowaniu przykłady (widoki) można również pobrać ze strony internetowej [www.isc.infor.pl](http://www.isc.infor.pl) (zakładka „Excel”).



**87 000**  
orzeczeń  
sądów i administracji  
[www.biznes2.inforlex.pl](http://www.biznes2.inforlex.pl)

**INFORLEX.PL**

**Umów się na bezpłatną prezentację serwisu:**  
(0-22) 761 32 99 | [inforlex@inforlex.pl](mailto:inforlex@inforlex.pl)