

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Excel. Funkcje w przykładach

Autor: Krzysztof Maślowski

ISBN: 83-246-0026-4

Format: B5, stron: 416



Arkusze kalkulacyjne Excel to jedno z najczęściej wykorzystywanych narzędzi informatycznych. Z jego możliwości korzystają nie tylko księgowi, finansiści i handlowcy, ale także naukowcy, którzy stosują go do przeprowadzania złożonych obliczeń i symulacji. Swoją potęgę Excel pokazuje dopiero wtedy, gdy użytkownik sięgnie po funkcje – gotowe mechanizmy obliczeniowe pozwalające na wykonanie niemal dowolnych działań i operacji obliczeniowych. Łącząc pojedyncze funkcje w formuły, można szybciej zrealizować praktycznie każde zadanie.

„Excel. Funkcje w przykładach” to książka przedstawiająca zasady stosowania funkcji w obliczeniach wykonywanych za pomocą arkusza kalkulacyjnego Excel. Czytając ją, poznasz podstawy tworzenia własnych formuł oraz nauczysz się reguł adresowania komórek i odwoływania się do nich w formułach. Dowiesz się również, jak przypisywać nazwy komórkom i zakresom oraz jak korzystać z nich podczas obliczeń. Znajdziesz tu także szczegółowe omówienie poszczególnych funkcji Excela.

- Kolejność operacji w formułach
- Wpisywanie i edycja formuł w komórkach
- Tworzenie i używanie nazw
- Funkcje logiczne
- Funkcje daty i czasu
- Funkcje informacyjne
- Funkcje wyszukiwania
- Funkcje matematyczne
- Funkcje bazy danych
- Funkcje statystyczne
- Funkcje finansowe

Kalkulatory już dawno stały się przeżytkiem.

Dziś najdoskonalszym narzędziem obliczeniowym jest Excel



Spis treści

Wstęp	13
Część I Budowanie warsztatu	15
Rozdział 1. Podstawy	17
Domyślna interpretacja wpisu do komórki	17
Operatory i kolejność operacji w formułach	21
Operatory odwołania	21
Operatory arytmetyczne	22
Operator konkatencji	23
Operatory porównania	23
Porównanie priorytetów operatorów używanych w formułach	25
Sprawdzanie kolejności wykonywania operacji w formułach	25
Dlaczego i do czego używamy funkcji? — przykłady	27
Obliczenia w komórkach arkusza	27
Formatowanie warunkowe	29
Sprawdzanie poprawności	31
Wprowadzanie formuł do komórek arkusza	32
Wpisywanie formuły z klawiatury	33
Wprowadzanie formuły przez wskazywanie adresów	33
Wprowadzanie funkcji do formuły	34
Edytowanie formuł	38
Wyświetlanie i znajdowanie formuł	39
Znajdowanie formuł po ich wyświetleniu	39
Znajdowanie formuł za pomocą okna dialogowego Przejdź do	39
Obliczenia automatyczne i ręczne	41
Odwołania do komórek w formułach	42
Odwołania względne	42
Odwołania bezwzględne	43
Adresy mieszane	44
Klawisz F4	45
Adresowanie w trzecim wymiarze	45
Błędy sygnalizowane przez formuły i funkcje	47
Rozdział 2. Nazwy	49
Zasady tworzenia nazw	50
Nazwy z odwołaniem bezwzględnym	50
Standardowe tworzenie nazw	50
Przypisywanie nazw do istniejących odwołań	51

Tworzenie nazw za pomocą paska formuły	52
Tworzenie nazw za pomocą tekstu wpisanego do sąsiednich komórek arkusza	52
Nazwy z odwołaniem względnym	53
Nazwy formuł	54
Nazwy stałych	55
Modyfikowanie i usuwanie nazw	56
Modyfikowanie nazwy	56
Usuwanie nazw	56
Nazwy na poziomie skoroszytu i nazwy na poziomie arkusza	57
Nadawanie nazw na poziomie arkusza	58
Analiza zależności między nazwami na poziomie arkusza i na poziomie skoroszytu	59
Praktyczny sposób budowania skoroszytów z nazwami na poziomie arkusza	60
Znajdowanie nazw	60
Wyszukiwanie na liście nazw definiowanych	61
Wyszukiwanie nazw za pomocą paska formuły	61
Znajdowanie nazw przez zmniejszanie skali wyświetlania	62
Wklejanie listy nazw	62
Znajdowanie nazwanych zakresów za pomocą okna Przechodzenie do	63
Menadżer nazw	64
Rzadko używany operator części wspólnej zakresów	65
Wskazówki kończące część I	66

Część II Funkcje 67

Rozdział 3. Funkcje logiczne 71

PRAWDA — TRUE	71
FAŁSZ — FALSE	72
NIE — NOT	72
JEŻELI — IF	75
ORAZ — AND	76
LUB — OR	78

Rozdział 4. Funkcje daty i czasu 83

CZAS — TIME	85
CZAS.WARTOŚĆ — TIMEVALUE	87
DATA — DATE	88
DATA.RÓŻNICA — DATEDIF	89
DATA.WARTOŚĆ — DATEVALUE	91
DNI.360 — DAYS360	92
DZIEŃ — DAY	94
DZIEŃ.TYG — WEEKDAY	95
DZIŚ — TODAY	97
EDATE	98
EOMONTH	101
GODZINA — HOUR	102
MIESIĄC — MONTH	105
MINUTA — MINUTE	106
NETWORKDAYS	107
ROK — YEAR	109
SEKUNDA — SECOND	110
TERAZ — NOW	111
WEEKNUM	113
WORKDAY	115
YEARFRAC	116

Rozdział 5. Funkcje informacyjne	117
BRAK — NA	117
CZY.ADR — ISREF	119
CZY.BL — ISERR	120
CZY.BŁĄD — ISERROR	122
CZY.BRAK — ISNA	123
CZY.LICZBA — ISNUMBER	124
CZY.LOGICZNA — ISLOGICAL	126
CZY.NIE.TEKST — ISNONTEXT	127
CZY.PUSTA — ISBLANK	128
CZY.TEKST — ISTEXT	129
INFO	131
ISEVEN	133
ISODD — ISODD	134
KOMÓRKA — CELL	135
L — N	139
NR.BŁĘDU — ERROR.TYPE	140
TYP — TYPE	141
Rozdział 6. Funkcje tekstowe	143
ASC	143
BAHTTEXT	144
DŁ — LEN	145
FRAGMENT.TEKSTU — MID	146
KOD — CODE	148
KWOTA — DOLLAR	149
LEWY — LEFT	150
LITERY.MAŁE — LOWER	151
LITERY.WIELKIE — UPPER	152
OCZYŚĆ — CLEAN	153
PODSTAW — SUBSTITUTE	153
PORÓWNAJ — EXACT	155
POWT — REPT	156
PRAWY — RIGHT	157
SZUKAJ.TEKST — SEARCH	157
T — T	159
TEKST — TEXT	160
USUŃ.ZBĘDNE.ODSTĘPY — TRIM	162
WARTOŚĆ — VALUE	163
Z.WIELKIEJ.LITERY — PROPER	164
ZAKR.DO.TEKST — FIXED	164
ZASTĄP — REPLACE	166
ZŁĄCZ.TEKSTY — CONCATENATE	168
ZNAJDŹ — FIND	168
ZNAK — CHAR	170
Rozdział 7. Funkcje wyszukiwania i adresu	171
ADR.POŚR — INDIRECT	171
ADRES — ADDRESS	174
HIPERŁĄCZE — HYPERLINK	176
ILE.WIERSZY — ROWS	178
INDEKS — INDEX	179
LICZBA.KOLUMN — COLUMNS	182

NR.KOLUMNY — COLUMN	184
OBSZARY — AREAS	185
PODAJ.POZYCJĘ — MATCH	185
PRZESUNIĘCIE — OFFSET	191
TRANSPONUJ — TRANSPOSE	195
WIERSZ — ROW	196
WYBIERZ — CHOOSE	197
WYSZUKAJ — LOOKUP	199
WYSZUKAJ.PIONOWO — VLOOKUP	201
WYSZUKAJ.POZIOMO — HLOOKUP	202
Rozdział 8. Funkcje matematyczne	205
ACOS — ACOS	205
ACOSH — ACOSH	206
ASIN — ASIN	206
ASINH — ASINH	207
ATAN — ATAN	207
ATAN2 — ATAN2	207
ATANH — ATANH	208
COS — COS	209
COSH — COSH	210
EXP — EXP	211
FACTDOUBLE — FACTDOUBLE	211
GCD — GCD	212
ILOCZYN — PRODUKT	212
KOMBINACJE — COMBIN	213
LCM — LCM	213
LICZBA.CAŁK — TRUNC	214
LN — LN	216
LOG — LOG	217
LOG10 — LOG10	218
LOS — RAND	218
MACIERZ.ILOCZYN — MMULT	219
MACIERZ.ODW — MINVERSE	220
MOD — MOD	222
MODUŁ.LICZBY — ABS	226
MROUND — MROUND	227
MULTINOMIAL — MULTINOMIAL	228
PI — PI	229
PIERWIASTEK — SQRT	233
POTEGA — POWER	234
QUOTIENT — QUOTIENT	235
RADIANY — RADIANS	236
RANDBETWEEN — RANDBETWEEN	236
RZYMSKIE — ROMAN	238
SERIESSUM — SERIESSUM	239
SILNIA — FACT	240
SIN — SIN	240
SINH — SINH	241
SQRTPI — SQRTPI	241
STOPNIE — DEGREES	241
SUMA — SUM	242

SUMA.ILOCZYNÓW — SUMPRODUCT	244
SUMA.JEŻELI — SUMIF	246
SUMA.KWADRATÓW — SUMSQ	248
SUMA.X2.M.Y2 — SUMX2MY2	249
SUMA.X2.P.Y2 — SUMX2PY2	250
SUMA.XMY.2 — SUMXMY2	251
SUMY.POŚREDNIE — SUBTOTAL	252
TAN — TAN	260
TANH — TANH	260
WYZNACZNIK.MACIERZY — MDETERM	261
ZAKR — ROUND	263
ZAKR.DO.CAŁK — INT	264
ZAKR.DO.NPARZ — ODD	265
ZAKR.DO.PARZ — EVEN	265
ZAKR.DÓŁ — ROUNDDOWN	266
ZAKR.GÓRA — ROUNDUP	267
ZAKR.W.DÓŁ — FLOOR	267
ZAKR.W.GÓRĘ — CEILING	268
ZNAK.LICZBY — SIGN	269
Rozdział 9. Funkcje bazy danych	273
Kryteria	274
Ogólne zasady tworzenia kryteriów	274
Koniunkcja i alternatywa warunków	275
Warunek w postaci formuły	278
Równa się czy zaczyna się od?	279
Znaki globalne	280
BD.ILE.REKORDÓW — DCOUNT	281
BD.ILE.REKORDÓW.A — DCOUNTA	282
BD.ILOCZYN — DPRODUCT	284
BD.MAX — DMAX	286
BD.MIN — DMIN	287
BD.ODCH.STANDARD — DSTDEV	288
BD.ODCH.STANDARD.POPUL — DSTDEVP	289
BD.POLE — DGET	289
BD.SUMA — DSUM	290
BD.ŚREDNIA — DAVERAGE	291
BD.WARIANCJA — DVAR	293
BD.WARIANCJA.POPUL — DVARP	294
Rozdział 10. Funkcje statystyczne	295
CZĘSTOŚĆ — FREQUENCY	296
ILE.LICZB — COUNT	299
ILE.NIEPUSTYCH — COUNTA	301
KOWARIANCJA — COVAR	303
KURTOZA — KURT	306
KWARTYL — QUARTILE	307
LICZ.JEŻELI — COUNTIF	308
LICZ.PUSTE — COUNTBLANK	312
MAX — MAX	313
MAX.A — MAXA	315
MAX.K — LARGE	315

MEDIANA — MEDIAN	317
MIN — MIN	319
MIN.A — MINA	320
MIN.K — SMALL	320
NACHYLENIE — SLOPE	322
NORMALIZUJ — STANDARDIZE	323
ODCH.KWADRATOWE — DEVSQ	326
ODCH.STANDARD.POPUL — STDEVP	327
ODCH.STANDARD.POPUL.A — STDEVPA	328
ODCH.STANDARDOWE — STDEV	329
ODCH.STANDARDOWE.A — STDEVA	330
ODCH.ŚREDNIE — AVEDEV	330
ODCIĘTA — INTERCEPT	331
PEARSON — PEARSON	332
PERCENTYL — PERCENTILE	333
Trochę teorii	334
PERMUTACJE — PERMUT	335
Trochę teorii	336
POZYCJA — RANK	338
PRAWDP — PROB	340
PROCENT.POZYCJA — PERCENTRANK	341
PRÓG.ROZKŁAD.DWUM — CRITBINOM	343
R.KWADRAT — RSQ	344
Informacje ogólne o regresji. Funkcje regresji nieliniowej	345
REGLINP — LINEST	345
REGLINW — TREND	347
REGLINX — FORECAST	349
ROZKŁAD.NORMALNY — NORMDIST	353
SKOŚNOŚĆ — SKEW	356
ŚREDNIA — AVERAGE	359
ŚREDNIA.A — AVERAGEA	360
ŚREDNIA.GEOMETRYCZNA — GEOMEAN	360
ŚREDNIA.HARMONICZNA — HARMEAN	361
ŚREDNIA.WEWN — TRIMMEAN	363
TEST.CHI — CHITEST	363
TEST.F — FTEST	364
TEST.T — TTEST	364
TEST.Z — ZTEST	365
UFNOŚĆ — CONFIDENCE	365
WARIANCJA — VAR	366
WARIANCJA.A — VARA	366
WARIANCJA.POPUL — VARP	367
WARIANCJA.POPUL.A — VARPA	367
WSP.KORELACJI — CORREL	368
WYST.NAJCZĘŚCIEJ — MODE	369
Rozdział 11. Funkcje finansowe	371
Wbudowane funkcje finansowe	372
DB	372
DDB	374
FV	374
IPMT	376
IRR	378

ISPMT	379
MIRR	380
NPER	381
NPV	382
PMT	384
PPMT	385
PV	385
RATE	386
SLN	388
SYD	389
VDB	389
Funkcje finansowe z pakietu ATP	390
Rozdział 12. Funkcje inżynierskie	395
Skorowidz	399

Rozdział 6.

Funkcje tekstowe

Funkcje tekstowe służą w zasadzie do działań na tekstach, ale Excel jest tu tolerancyjny i często argumentami mogą być także liczby. Przekonamy się o tym, analizując podane w tym rozdziale przykłady. Dzięki funkcjom tekstowym można łatwo między innymi: wyszukiwać w tekście wybrane znaki, zamieniać je na inne, pozbywać się zbędnych spacji lub zamieniać małe litery na wielkie bądź odwrotnie.

Nie wszystkie funkcje działające na tekstach lub tekstów dotyczące zostały włączone do tej kategorii. Na przykład, opisana w poprzednim rozdziale, funkcja CZY.TEKST została zaliczona do funkcji informacyjnych. Również w innych kategoriach znajdują się funkcje działające na tekstach, lecz o nich, zgodnie z porządkiem tej książki, będzie mowa w innych rozdziałach.

Podstawowe funkcje tekstowe obsługują teksty, w których znaki są pojedynczymi bajtami. Funkcje te mają swoje odpowiedniki przeznaczone do obsługi tekstów pisanych znakami dwubajtowymi (znakami pełnej szerokości), co ma miejsce w takich językach jak tajski. Ich nazwy są rozszerzone o kropkę oddzielającą i literę B. Na przykład odpowiednikiem funkcji DŁ jest funkcja DŁ.B. Pierwsza zwraca liczbę znaków jednobajtowych w tekście, a druga liczbę bajtów w tekstach zapisanych znakami dwubajtowymi. Ponieważ przydatność funkcji obsługujących teksty zapisane znakami dwubajtowymi jest w Polsce niewielka, nie będę ich opisywał. Opiszę jedynie kilka funkcji napisanych specjalnie dla języków ze znakami dwubajtowymi i nie mających odpowiedników obsługujących zwykłe teksty. Przykładem takiej funkcji jest np. BAHTTEXT.

ASC

Funkcja wbudowana, dostępna w VBA.

W językach z dwubajtowym zestawem znaków (DBCS) zmienia znaki o pełnej szerokości (dwubajtowe) na znaki o połówkowej szerokości (jednobajtowe).

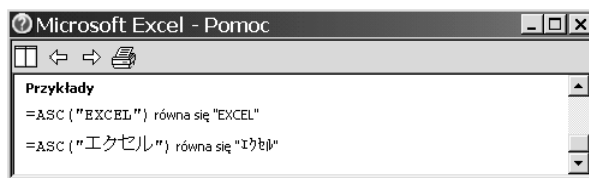
Składnia:

=ASC(
ASC(tekst)	

- ♦ tekst — to tekst lub odwołanie do komórki zawierającej tekst. Tekst jest zmieniany tylko wtedy, gdy zawiera znaki dwubajtowe — patrz rysunek 6.1. W przykładzie podanym w systemie pomocy Excel, słowo „EXCEL”, nie zawierające znaków dwubajtowych, nie zostało zmienione.

Rysunek 6.1.

Działanie funkcji ASC



W VBA istnieje funkcja Asc zwracająca kod ASCII pierwszego znaku tekstu. Funkcja VBA jest używana przez bezpośrednie użycie w procedurze; wywołanie arkuszowej funkcji ASC musi być zgodne z metodą wywoływania funkcji arkuszowych, opisaną we wstępie do części II.

BAHTTEXT

Funkcja wbudowana, dostępna w VBA.

Zamienia liczbę na tekst w języku tajskim, dodaje przyrostek „Baht”.

Składnia:

=BAHTTEXT(
BAHTTEXT(liczba)	

- ♦ liczba — to liczba lub odwołanie do komórki zawierającej liczbę, co pokazuje przykład z systemu pomocy Excela pokazany na rysunku 6.2.

Rysunek 6.2.

*Działanie funkcji
BAHTTEXT*

	A	B	C
1	1234		
2	หนึ่งพันสองร้อยสามสิบบาทถ้วน	<--	=BAHTTEXT(A1)
3			

Ciekawe, że odpowiedniej funkcji nie ma dla żadnego innego języka, nawet dla angielskiego. John Walkenbach, jeden z najlepszych znawców Excela, przypuszcza, że w grupie programistów tworzących ten program jest wielu miłośników kuchni tajskiej, którzy w ten sposób wyrazili swą wdzięczność.

DŁ — LEN

Funkcja wbudowana, niedostępna w VBA.

Oblicza liczbę znaków ciągu tekstowego.

Składnia:



- ♦ tekst — tekst lub odwołanie do komórki zawierającej tekst.

Przykład 6.1. Wyróżnianie zbędnych spacji

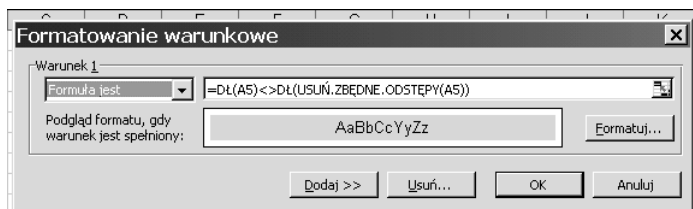
Plagą wielu list, zwłaszcza otrzymywanych ze źródeł zewnętrznych, są dodatkowe spacje; utrudniają np. sortowanie i grupowanie wyników, tworzenie sum pośrednich itd., bowiem dla Excela Jan Kowalski z jedną spacją między imieniem i nazwiskiem to całkiem inna osoba niż Jan Kowalski z dwiema spacjami. Najtrudniejsze do wyszukania są komórki zawierające dodatkowe spacje na końcu tekstu, gdyż takich spacji nie widzimy przy przeglądaniu arkusza (np. w komórce A5 na rysunku 6.3)¹. Ale wyszukanie nawet widocznych spacji nadmiarowych nie jest możliwe, jeżeli danych są tysiące.

Należy pamiętać, że programy baz danych nie tolerują takich niedokładności i przed eksportem przygotowanej listy do pliku, który ma być czytany przez inne oprogramowanie, należy dane bezwzględnie oczyścić ze wszystkich śmieci, w tym z dodatkowych spacji. Łatwiej to zrobić, jeżeli komórki z dodatkowymi spacjami wyróżnimy za pomocą formatowania warunkowego, tak jak na rysunku 6.3.

Rysunek 6.3.

Użycie formatowania warunkowego pozwala wyróżnić komórki, w których tekst zawiera dodatkowe spacje

	A	B		A	B
1	Osoba	Wpłata	1	Osoba	Wpłata
2	Jan Kowalski	125,00 zł	2	Jan Kowalski	125,00 zł
3	Anna Giza	23,66 zł	3	Anna Giza	23,66 zł
4	Marek Nowak	254,20 zł	4	Marek Nowak	254,20 zł
5	Anna Giza	345,12 zł	5	Anna Giza	345,12 zł
6	Jacek Gocz	45,00 zł	6	Jacek Gocz	45,00 zł
7	Paweł Wilk	124,21 zł	7	Paweł Wilk	124,21 zł
8	Jan Kowalski	73,61 zł	8	Jan Kowalski	73,61 zł
9	Artur Kolski	341,00 zł	9	Artur Kolski	341,00 zł



¹ Excel takie końcowe spacje traktuje dość tolerancyjnie i pomija np. przy automatycznym tworzeniu sum pośrednich, ale dla spacji w środku tekstu nie jest wyrozumiały.

Zauważ, że w formule warunku formatowania zostały użyte adresy względne. W komórce A5 tekst Anna Giza został wyróżniony z powodu dodatkowej spacji na końcu, widocznej po ustawieniu kursora po przejściu do edycji zawartości komórki.

Tekst jest wyróżniany, jeżeli jego długość po usunięciu zbędnych spacji (w wyniku działania funkcji USUŃ.ZBĘDNE.ODSTĘPY) jest różna od pierwotnej.

FRAGMENT.TEKSTU — MID

Funkcja wbudowana, niedostępna w VBA.

Zwraca fragment ciągu tekstowego — określoną liczbę znaków, począwszy od podanej pozycji.

Składnia:

=FRAGMENT.TEKSTU(FRAGMENT.TEKSTU(tekst ; liczba_początkowa; liczba_znaków)
--

- ♦ tekst — tekst lub odwołanie do komórki z tekstem zawierającym znaki, które należy wyodrębnić. Excel zezwala tu na użycie liczby jako pierwszego argumentu funkcji, ale zawsze uwzględnia jedynie jej rzeczywistą wartość, a nie postać wynikającą z formatowania (patrz rysunek 6.4).

Rysunek 6.4.

Funkcja FRAGMENT.TEKSTU zawsze używa rzeczywistej wartości liczby, nie uwzględniając formatowania

	C1	=FRAGMENT.TEKSTU(B1;4;2)			
	A	B	C	D	E
1	data	2006-05-19	56		
2	jej liczba kolejna	38856	56		
3					

- ♦ liczba_początkowa — określa pozycję pierwszego znaku, który należy wyodrębnić (numeracja rozpoczyna się od 1).
- ♦ liczba_znaków — liczba znaków, które należy wyodrębnić.

Jeżeli:	funkcja zwraca
liczba_początkowa jest > od długości tekstu	wartość "" (tekst pusty)
liczba_początkowa jest < od długości tekstu, ale liczba_początkowa plus liczba_znaków jest ≥ od długości tekstu	znaki do końca tekstu
liczba_początkowa jest < niż 1	wartość błędu #ARG!
liczba_znaków jest < 0	wartość błędu #ARG!

Przykład 6.2. Wybieranie cyfr z liczby

Jako pierwszego argumentu funkcji FRAGMENT.TEKSTU można użyć liczby lub odwołania do komórki zawierającej liczbę, jednakże otrzymany wówczas wynik może czasem zdziwić niedoświadczonego użytkownika Excela. Przykład został pokazany na rysunku 6.4.

W obu komórkach *B1* i *B2* jest zapisana ta sama liczba, którą w komórce *B2* pozostawiono w formacie ogólnym, zaś w komórce *B1* nadano jej format daty. Widać, że funkcja `FRAGMENT.TEKSTU` tego nie uwzględnia, zwracając w obu przypadkach ciąg tekstowy (zauważ wyrównanie do lewej) złożony z 4. i 5. cyfry pobranej z zapisu ogólnego.

Przykład 6.3. Konwersja nietypowego formatu daty

Często Excela używamy do obróbki danych otrzymywanych z zewnętrznych źródeł, np. z systemów komputerowych typu mainframe. Daty w takich plikach mogą być zapisane w różnych formatach niezrozumiałych dla Excela, co wymusza stosowanie konwersji.

Przykładem mogą być: data i czas zapisane w formacie `rrrrrdddggmm`, gdzie `rrrrr` oznacza rok, `ddd` kolejny dzień roku od 001 do 366, a `gg` i `mm` odpowiednio godziny i minuty w formacie 24-godzinnym.

Rysunek 6.5 pokazuje sposób poradzenia sobie z tym problemem.

Rysunek 6.5.

Użycie funkcji `FRAGMENT.TEKSTU` do konwersji daty i czasu zapisanych w niestandardowym formacie

	A	B	C	D	E	F	G
1	format rrrrrdddggmm	data i czas					
2	19992450112	03-09-1999 01:12					
3	19980342310	04-02-1998 23:10					
4	20053010702	29-10-2005 07:02					
5	20061641815	14-06-2006 18:15					
6	20062121705	01-08-2006 17:05					

```
=DATA(LEWY(A2;4);1;1)
+FRAGMENT.TEKSTU(A2;5;3)
+CZAS(FRAGMENT.TEKSTU(A2;8;2);
PRAWY(A2;2);0)
```

Jak widać na rysunku 6.5, całkowita formuła konwersji ma w komórce *B2* postać:

```
=DATA(LEWY(A2;4);1;1)
+FRAGMENT.TEKSTU(A2;5;3)
+CZAS(FRAGMENT.TEKSTU(A2;8;2);PRAWY(A2;2);0)
```

gdzie:

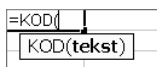
- ♦ `LEWY(A2;4)` pobiera cztery pierwsze cyfry (1999), co `DATA(LEWY(A2;4);1;1)` zamienia na 36161, czyli numer kolejny daty 1 stycznia 1999 roku;
- ♦ `FRAGMENT.TEKSTU(A2;5;3)` wycina z liczby zapisanej w *A2* trzy kolejne cyfry, począwszy od piątej, tworząc tekst "245";
- ♦ dzięki „domyślności” Excela dodanie liczby i tekstu 36161+"245" daje liczbę 36 406;
- ♦ `FRAGMENT.TEKSTU(A2;8;2)` daje tekst "01" złożony z 8. i 9. cyfry, a `PRAWY(A2;2)` tekst "12" złożony z cyfr 10. i 11., zaś `CZAS("01";"12";0)=CZAS(1;12;0)` daje kolejną liczbę czasu 0,05;
- ♦ `36 406+0,05=36 406,05` to kolejna liczba daty i czasu, która przez nadanie formatu `dd-mm-rrrr gg:mm` jest widoczna w postaci 03-09-1999 01:12, podając w zrozumiały sposób datę i godzinę.

KOD — CODE

Funkcja wbudowana, niedostępna w VBA.

Zwraca kod numeryczny pierwszego znaku ciągu tekstowego.

Składnia:



- ♦ tekst — tekst lub odwołanie do komórki zawierającej tekst.

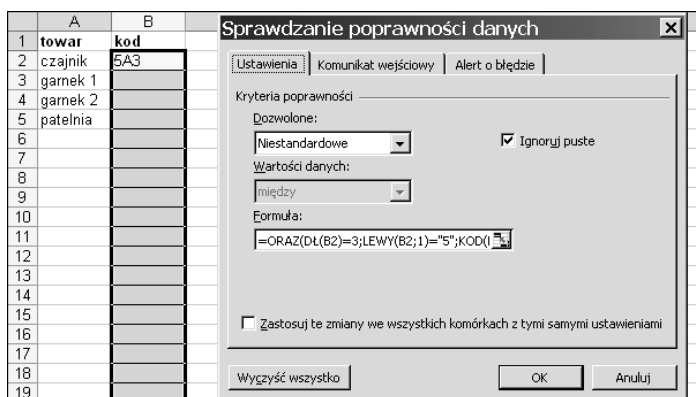
Przykład 6.4. Użycie funkcji KOD do sprawdzania poprawności wpisywanych danych

Często dane wpisywane do tabeli muszą spełniać określone warunki, być liczbami z odpowiedniego zakresu albo określonymi ciągami znaków. Załóżmy, że tworzymy tabelę: w pierwszej kolumnie znajduje się nazwa towaru, a w drugiej jej trzyznakowy kod, rozpoczynający się od cyfry 5, z wielką literą na drugim miejscu (bez liter polskich) i dowolną cyfrą na trzecim (np. 5A3 jest kodem prawidłowym, zaś 5ac, 6A3 i 5A33 nie spełniają założeń).

Należy zabezpieczyć się przed wpisywaniem błędnych kodów, co zrobimy za pomocą wbudowanej w Excela procedury sprawdzania poprawności. Funkcji KOD użyjemy do sprawdzenia, czy na drugim miejscu w ciągu znakowym znajduje się wielka litera. Dozwolone litery to A, B...Z. Ich kody to odpowiednio: 65, 66...90 — łatwo to sprawdzić, wpisując np. do B1 formułę =KOD(A1), a następnie do A1 litery, których kod chcemy odczytać. Polskie litery A, Ć itd. mają kody spoza tego zakresu i ich uwzględnienie wymagałoby rozbudowania, a więc wydłużenia formuły, a przy tym nie wniosłoby nic w istotę rozumienia działania funkcji KOD.

1. Zaznacz zakres, który ma być objęty sprawdzaniem — na rysunku 6.6 został on wyróżniony innym kolorem — i wydaj polecenie *Dane/Sprawdzanie poprawności*.

Rysunek 6.6.
Wpisywanie formuły sprawdzania poprawności kodu wpisywanego do kolumny B



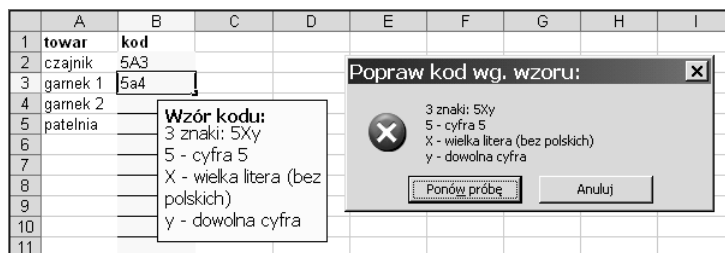
2. Na karcie *Ustawienia* wybierz *Dozwolone/Niestandardowe*, a w polu *Formuła* wpisz formułę:

```
=ORAZ(
DŁ(B2)=3;
LEWY(B2;1)="5";
KOD(FRAGMENT.TEKSTU(B2;2;1))>64;KOD(FRAGMENT.TEKSTU(B2;2;1))<91;
CZY.LICZBA(WARTOŚĆ(FRAGMENT.TEKSTU(B2;3;1)))
)
```

3. Na kartach *Komunikat wejściowy* i *Alert o błędzie* możesz dodatkowo wpisać komunikaty objaśniające, np. takie jak na rysunku 6.7.

Rysunek 6.7.

Próba wpisania błędnego kodu została oprotestowana



Elementy formuły sprawdzające poszczególne znaki kodu zostały w punkcie 2. zapisane w oddzielnych wierszach, aby ułatwić zrozumienie ich działania. Każdy pojedynczy element formuły jest prosty, więc pomijam szczegółowe objaśnienia.

KWOTA — DOLLAR

Funkcja wbudowana, dostępna w VBA.

Zmienia liczbę na tekst i dodaje symbol waluty, przy czym następuje zaokrąglenie do liczby miejsc dziesiętnych zgodnie z drugim argumentem funkcji. Stosowany format to # ##0,00 zł.

Składnia:

```
=KWOTA()
KWOTA(liczba; [miejsca_dziesiętne])
```

- ♦ *liczba* — liczba lub odwołanie do komórki zawierającej liczbę (ew. formułę zwracającą liczbę),
- ♦ *miejsca_dziesiętne* — liczba cyfr po przecinku dziesiętnym; pominięcie oznacza wartość 2. Dopuszczalne jest podawanie wartości ujemnych, których znaczenie pokazuje rysunek 6.8.

Rysunek 6.8.

Zaokrąglenia sterowane drugim argumentem funkcji KWOTA są wykonywane zgodnie z zasadami arytmetyki

	A	B	C	D
1	1234,5678			liczba źródłowa
2	1 234,57 zł	<--	=KWOTA(\$A\$1)	domyślnie dwa miejsca po przecinku (tekst)
3	1 234,57 zł	<--	=KWOTA(\$A\$1;2)	dwa miejsca po przecinku (tekst)
4	1 234,568 zł	<--	=KWOTA(\$A\$1;3)	trzy miejsca po przecinku (tekst)
5	1 235 zł	<--	=KWOTA(\$A\$1;0)	zaokrąglenie do jedności (tekst)
6	1 200 zł	<--	=KWOTA(\$A\$1;-2)	zaokrąglenie do setek (tekst)
7	1 234,57 zł	<--	=\$A\$1	liczba ze standardowym formatem walutowym
8				



Należy zwrócić uwagę na różnicę między działaniem funkcji KWOTA a nadawaniem liczbie formatu walutowego. Liczba, której nadano format walutowy, pozostaje liczbą (na rysunku 6.8 w komórce A7 — wyrównanie do prawej), zaś funkcja KWOTA zamienia liczbę na tekst (w komórkach A2:A6 — wyrównanie do lewej).

Przykład 6.5. Łączenie kwot walutowych z tekstem

Często musimy drukować różnego rodzaju zestawienia, w których tekst łączymy z sumami pieniężnymi. Nadanie liczbie formatu walutowego zmienia jedynie sposób jej wyświetlania, pozostawiając samą liczbę bez zmiany. Dlatego po połączeniu z tekstem znak waluty nie jest widoczny, jak w komórce C2 w górnej części rysunku 6.9. Potrzebne jest dołączenie dodatkowego członu tekstowego " zł", jak w komórce C3 na tym samym rysunku (formuły użyte w komórkach C2 i C3 są pokazane obok w kolumnie E).

Rysunek 6.9.

Porównanie rysunków górnego i dolnego pokazuje, że funkcja KWOTA ułatwia łączenie tekstu z sumami pieniężnymi

	A	B	C	D	E
1	osoba	zapłata	opis		
2	Florek	123,50 zł	Florek otrzymał 123,5	<--	=A2&" otrzymał "&B2
3	Gawin	23,44 zł	Gawin otrzymał 23,44 zł	<--	=A3&" otrzymał "&B3&" zł"

	A	B	C	D	E
1	osoba	zapłata	opis		
2	Florek	123,50 zł	Florek otrzymał 123,50 zł	<--	=A2&" otrzymał "&KWOTA(B2)
3	Gawin	23,44 zł	Gawin otrzymał 23,44 zł		

Dołączania dodatkowego członu, złożonego ze spacji i symbolu waluty, można uniknąć, jeżeli użyjemy funkcji KWOTA, jak w dolnej części rysunku 6.9 w komórkach C2 i C3 (użyta formuła jest widoczna w komórce E2).

LEWY — LEFT

Funkcja wbudowana, niedostępna w VBA.

Zwraca pierwszy znak lub pierwsze znaki ciągu tekstowego.

Składnia:

=LEWY(<input type="text"/>)			
LEWY(tekst; [liczba_znaków])			

- ♦ tekst — tekst (ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy) zawierający znaki, które należy wyodrębnić.

- ♦ `liczba_znaków` — określa, ile znaków ma zwrócić funkcja LEWY. Liczba ta musi być większa lub równa zero. Jeśli `liczba_znaków` jest większa niż długość tekstu, wynikiem funkcji LEWY jest cały tekst. Pominięcie argumentu `liczba_znaków` jest równoznaczne z podaniem wartości 1.

Możliwości użycia funkcji LEWY zostały zaprezentowane w przykładach 6.3, 6.4, 6.6, 6.7 i 6.26. Jest często stosowana razem z funkcjami FRAGMENT.TEKSTU i ZNAJDŹ.

LITERY.MAŁE — LOWER

Funkcja wbudowana, niedostępna w VBA.

Zmienia w tekście litery wielkie na małe. Innych znaków nie zmienia.

Składnia:

=LITERY.MAŁE(
LITERY.MAŁE(tekst)	

- ♦ `tekst` — tekst (ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy).

Przykład 6.6. Tworzenie inicjałów pisanych małymi literami

Excel jest często używany jako narzędzie przygotowywania danych eksportowanych później do baz danych, systemów księgowych itd. Niektóre takie systemy odróżniają litery wielkie i małe i dla nich Kowalski i kowalski to dwie różne osoby, a e1JK i e1jk to różne symbole. Dlatego w danych eksportowanych używanie liter wielkich bądź małych powinno być ściśle kontrolowane.

Na podstawie listy pracowników i działów, w których pracują, należy utworzyć kody pracowników. Każdy kod ma być pisany małymi literami i składać się z dwuznakowego oznaczenia działu, pierwszej litery imienia i trzech pierwszych liter nazwiska. Rozwiązanie zostało pokazane na rysunku 6.10.

Rysunek 6.10.

Tworzenie kodów pracowniczych pisanych małymi literami

	A	B	C	D	E	F
1	Imię	Nazwisko	Dział	Kod pracownika		
2	Sebastian	Florek	a1	a1sflo		
3	Andrzej	Gawin	a2	a2agaw		
4	Jan	Kowalski	k1	k1jkow		
5	Marek	Nowak	k2	k2mnow		

Działanie formuły pokazanej na pasku edycji na rysunku 6.10 jest tak proste, że nie będę go objaśniał.



W kodach użyto trzech liter nazwiska, aby uniknąć powtarzania się kodów. Używanie jedynie pierwszych liter imienia i nazwiska zwykle prowadzi do konfliktów, gdyż często w jednej firmie, a nawet w jednym dziale pracują osoby, mające te same inicjały, np. Jan Kowalski i Joanna Kącka.

LITERY.WIELKIE — UPPER

Funkcja wbudowana, niedostępna w VBA.

Zmienia w tekście litery małe na wielkie. Innych znaków nie zmienia.

Składnia:

=LITERY.WIELKIE(
LITERY.WIELKIE(tekst)	

- ♦ tekst — tekst (ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy).

Przykład 6.7. Sprawdzanie wprowadzonych danych. Modyfikacja przykładu 6.4

W przykładzie 6.4 tworzyliśmy tabelę, której pierwsza kolumna zawierała nazwę towaru, a drugiej — jej trzyznakowy kod, rozpoczynający się od cyfry 5, z wielką literą na drugim miejscu (bez polskich liter) i dowolną cyfrą na trzecim (np. 5A3 jest kodem prawidłowym, zaś 5ac, 6A3 i 5A33 nie spełniają założeń).

Teraz chcemy zezwolić, aby drugim znakiem kodu była po prostu litera, obojętnie, czy wielka, czy mała. Wymaga to zmiany formuły sprawdzającej dane na następującą:

```
=ORAZ(
DŁ(B2)=3;LEWY(B2:1)="5";
KOD(LITERY.WIELKIE(FRAGMENT.TEKSTU(B2:2;1)))>64;KOD(LITERY.WIELKIE
(FRAGMENT.TEKSTU(B2:2;1)))<91;
CZY.LICZBA(WARTOŚĆ(FRAGMENT.TEKSTU(B2:3;1)))
)
```

Formuła `FRAGMENT.TEKSTU(B2:2;1)` „wykroi” z komórki drugi wpisany znak. Może to być litera mała lub jakiś inny znak (w tym litera wielka). W pierwszym przypadku funkcja `LITERY.WIELKIE` zamieni małą literę na wielką, w drugim nie nastąpi żadna zmiana. Kody liter wielkich należą do przedziału od 65 (kod A) do 90 (kod Z), więc sprawdzenie, czy kod znaku zwróconego przez funkcję `LITERY.WIELKIE` jest > 64 oraz < 91 , jest równoznaczne ze sprawdzeniem, czy znakiem wpisanym na drugim miejscu jest litera (niezależnie od tego, czy jest mała, czy wielka).

Oczywiście, jak zwykle w Excelu, ta formuła mogła być napisana w inny sposób, ale to pozostawiam domyślności czytelników.

OCZYŚĆ — CLEAN

Funkcja wbudowana, dostępna w VBA.

Usuwa z tekstu wszystkie znaki niedrukowane.

Składnia:

=OCZYŚĆ(
OCZYŚĆ(tekst)		

- ♦ tekst — tekst (ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy).

Przykład 6.8. Oczyszczanie importowanych danych

W danych importowanych ze źródeł zewnętrznych, np. z systemów mainframe lub z Internetu, trafiają się znaki niedrukowalne (nierozpoznawalne), np. znaki o kodach z zakresu 1 – 31 oraz 129, 141, 143, 144 i 157. Często kilka takich różnych znaków Excel wyświetla w arkuszu tak samo — np. znaki o kodach 11 i 129 są pokazywane jako kwadraciki, widoczne na rysunku 6.11.

Rysunek 6.11.

Funkcja OCZYŚĆ usuwa znaki o „złych” kodach, tworząc zbitki, czasem wymagające rozdzielenia, jak na prawej części tego rysunku

	B1		f _x =OCZYŚĆ(A1)		B1		f _x =OCZYŚĆ(A1)
	A	B	C	D	A	B	C
1	12□46	1246			1	Jan□Kowalski	JanKowalski
2	67□19	6719			2	Wanda□Olej	WandaOlej
3	94□35	9435			3	Marek□Trzos	MarekTrzos
4	11□63	1163			4	Anna□Warska	AnnaWarska

PODSTAW — SUBSTITUTE

Funkcja wbudowana, dostępna w VBA.

Zamienia w ciągu tekstowym stary_tekst na nowy_tekst.

Składnia:

=PODSTAW(
PODSTAW(tekst; stary_tekst; nowy_tekst; [wystapienie_liczba])					

- ♦ tekst — tekst (ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy),
- ♦ stary_tekst — to tekst lub odwołanie do komórki zawierającej tekst, którego część znaków należy zastąpić,
- ♦ nowy_tekst — to tekst, który ma zastąpić stary_tekst,

- ♦ `wystapienie_liczba` — jeżeli `stary_tekst` występuje w tekście kilkakrotnie, `wystapienie_liczba` określa, przy którym wystąpieniu dojdzie do zamiany. Jeżeli ten argument jest pominięty, każdy znaleziony `stary_tekst` zostanie zamieniony na `nowy_tekst`.

Funkcji `PODSTAW` należy używać, gdy określony ciąg znaków ma być zamieniony na inny, np. formuła `=PODSTAW("mama";"m";"t")` zamienia `m` na `t` i daje tekst `tata`, przy czym miejsce wystąpienia `m` nie ma znaczenia.

Funkcja `ZASTĄP` służy do zamiany na inny dowolnego tekstu występującego w określonym miejscu ciągu tekstowego, np. formuła `=ZASTĄP("mama";3;2;"terac")` dwa znaki tekstu, zaczynając od trzeciego, zamienia na nowy tekst `terac`, przy czym istotne jest miejsce rozpoczęcia zmiany i liczba zmienianych znaków, a nie ma znaczenia to, jakie to są znaki.

Przykład 6.9. Zliczanie wystąpień wybranego znaku

Czasami chcemy sprawdzić, ilokrotnie wybrany znak występuje w tekście zapisanym w komórce. Rysunek 6.12 przedstawia rozwiązanie.

Rysunek 6.12.

Zliczanie wystąpień liter w tekście

	A	B	C	D
1	Zliczanie wystąpień wybranego znaku			
2	1 <-	=DŁ(\$A\$1)-DŁ(PODSTAW(\$A\$1;"Z";""))		tylko wielkie litery "Z"
3	2 <-	=DŁ(\$A\$1)-DŁ(PODSTAW(\$A\$1;"z";""))		tylko małe litery "z"
4	3 <-	=DŁ(\$A\$1)-DŁ(PODSTAW(LITERY.WIELKIE(\$A\$1);"Z";""))		wielkie i małe litery razem
5	3 <-	=DŁ(\$A\$1)-DŁ(PODSTAW(LITERY.MAŁE(\$A\$1);"z";""))		wielkie i małe litery razem
6				

Funkcja `PODSTAW` zamienia wybrany znak na tekst pusty, czyli praktycznie go usuwa i zwraca tekst skrócony o tyle, ile wystąpień danego znaku zostało znalezionych. Odjęcie długości tekstu skróconego od długości tekstu pełnego informuje, ile razy dany znak został w tekście znaleziony.

Ponieważ funkcja `podstaw` odróżnia litery wielkie i małe, formuła w komórce `A2` zliczyła jedynie wystąpienia wielkiej litery `Z`, a formuła w `A3` jedynie wystąpienia małej litery `z`.

W zmienionej formule w komórce `A4`

```
=DŁ($A$1)-DŁ(PODSTAW(LITERY.WIELKIE($A$1);"Z";""))
```

najpierw funkcja `LITERY.WIELKIE` zamienia wszystkie litery na wielkie, po czym wyszukuje w tekście wielką literę `Z`. W tak zmienionym tekście wielkie litery `Z` są zamieniane na pusty tekst. Różnica długości tekstu pierwotnego i skróconego odpowiada liczbie znalezionych liter `z` (razem wielkich i małych).

Formuła w `A5` działa analogicznie; również zlicza wszystkie litery `z` (wielkie i małe), dokonując przed podstawieniem tekstu pustego zamiany wszystkich liter na małe.

PORÓWNAJ — EXACT

Funkcja wbudowana, niedostępna w VBA.

Porównuje dwa teksty i zwraca wartość PRAWDA, jeśli są takie same, lub FAŁSZ, gdy są różne. Uwzględnia wielkość liter, ale ignoruje różnice w formatowaniu (patrz rysunek 6.13).

Rysunek 6.13.

Porównanie bezpośrednie komórek a użycie funkcji PORÓWNAJ

	A	B	C	D	E	F	G
1							Porównanie bezpośrednie
2	Styczeń	STYCZEŃ	PRAWDA	<--	=A1=B1		nie odróżnia liter wielkich od małych
3							
4							Funkcja PORÓWNAJ
5	Styczeń	STYCZEŃ	FAŁSZ	<--	=PORÓWNAJ(A2;B2)		odróżnia litery wielkie od małych
6	Styczeń	Styczeń	PRAWDA	<--	=PORÓWNAJ(A3;B3)		nie odróżnia formatowania
7							

Składnia:

=PORÓWNAJ(
	PORÓWNAJ(tekst1; tekst2)			

- ♦ tekst1, tekst2 — porównywane teksty lub odwołania do komórek zawierających teksty (ciągi tekstowe).

Przykład 6.10. Zliczanie tekstów pisanych małymi (wielkimi literami)

W kolumnie A mamy zapisane kody jakichś operacji. Są to ciągi tekstowe złożone z liter i cyfr. W kodach operacji przedpołudniowych użyto liter małych, a w kodach operacji popołudniowych — liter wielkich. Naszym zadaniem jest policzenie operacji przedpołudniowych i popołudniowych. Rozwiązanie zostało pokazane na rysunku 6.14.

Rysunek 6.14.

Użycie funkcji PORÓWNAJ do zliczania tekstów pisanych małymi bądź wielkimi literami

	A	B	C	D	E	F	G	H	I
1	Kod operacji	Przed pld.	Po pld.						
2	ab14	1	0	<--	=JEŻELI(PORÓWNAJ(LITERY.MAŁE(A2);A2);1;0)				
3	AX8	0	1						
4	ZY123	0	1						
5	SS14V	0	1						
6	nx653	1	0						
7	Razem	2	3						
8									

Formuła użyta w kolumnie B (pokazana na rysunku 6.14 na pasku edycji)

=JEŻELI(PORÓWNAJ(LITERY.MAŁE(A2);A2);1;0)

daje 1, gdy wszystkie litery występujące w tekście są małe.

Jeżeli w tekście w komórce A2:

- ♦ nie ma liter wielkich, oba teksty LITERY.MAŁE(A2) i A2 będą identyczne, więc formuła PORÓWNAJ(LITERY.MAŁE(A2);A2) zwróci wartość PRAWDA — wówczas funkcja JEŻELI zwróci wartość drugiego argumentu, czyli 1;

- ♦ są litery wielkie, to teksty LITERY.MAŁE(A2) i A2 będą różne i formuła PORÓWNAJ(LITERY.MAŁE(A2):A2) zwróci wartość FAŁSZ, a funkcja JEŻELI wartość trzeciego argumentu, czyli 0.

W analogiczny sposób można przeanalizować działanie formuły

=JEŻELI(PORÓWNAJ(LITERY.WIELKIE(A2):A2):1:0)

użytej w kolumnie C do zliczania tekstów zapisanych wielkimi literami.

POWT — REPT

Funkcja wbudowana, dostępna w VBA.

Tworzy ciąg tekstowy, powtarzając podany tekst określoną liczbę razy.

Składnia:

=POWT(
POWT(tekst; ile_razy)		

- ♦ tekst — tekst, który ma być powtarzany,
- ♦ ile_razy — liczba dodatnia określająca liczbę powtórzeń tekstu.

Jeśli argument ile_razy

- ♦ ma wartość 0, funkcja POWT zwraca "" (pusty tekst),
- ♦ nie jest liczbą całkowitą, jest do takiej liczby obcinany².

Wynik funkcji POWT nie może być dłuższy niż 32 767 znaków. W przeciwnym wypadku funkcja zwróci wartość błędu #ARG!.

Przykład 6.11. Tworzenie prostego histogramu

Funkcji POWT można użyć do prostej graficznej prezentacji danych, co zostało pokazane na rysunku 6.15.

Rysunek 6.15.
Użycie funkcji POWT do prostej graficznej prezentacji danych

	C2		fx =POWT("x";A2)				
	A	B	C	D	E	F	G
1	Wartość						
2	4		xxxx				
3	1		x				
4	20		xxxxxxxxxxxxxxxxxxxxxxxx				
5	26		xxxxxxxxxxxxxxxxxxxxxxxx				
6	11		xxxxxxxxxxx				
7	9		xxxxxxxxx				
8							

² Obcinany, a nie zaokrąglany zgodnie z regułami arytmetyki, np. ile_razy = 3,7 spowoduje trzykrotne powtórzenie tekstu (przy zaokrągleniu byłyby to 4 powtórzenia).

PRAWY — RIGHT

Funkcja wbudowana, niedostępna w VBA.

Zwraca ostatni znak lub znaki wyodrębnione z ciągu tekstowego.

Składnia:

=PRAWY(
PRAWY	(tekst;	[liczba_znaków])	

- ♦ tekst — ciąg tekstowy lub odwołanie do komórki zawierającej ciąg tekstowy, zawierający znaki, które mają być wyodrębnione.
- ♦ liczba_znaków — liczba znaków, jakie ma zwrócić funkcja PRAWY. Argument liczba_znaków musi być większy lub równy zero. Argument opcjonalny domyślnie równy 1.

Jeśli argument liczba_znaków jest większy od długości tekstu, to funkcja PRAWY zwraca cały tekst.

O zastosowaniu funkcji PRAWY była już mowa w przykładzie 4.3.