

INFORMATYKA I MATEMATYKA

Michał Bernardelli, Mariusz Kozakiewicz

ALGORYTMY KONWERSJI WARTOŚCI LICZBOWYCH NA ODPOWIEDNIKI TEKSTOWE W JĘZYKACH NARODOWYCH

[**słowa kluczowe:** arkusz kalkulacyjny, Open Office Calc, makro, makropolecenie, programowanie, Visual Basic for Applications, algorytm, zastosowania komputerów]

Streszczenie

Open Office Calc jest jednym z najczęściej używanych, darmowych arkuszy kalkulacyjnych na świecie. Umożliwia on użytkownikom rozszerzenie funkcjonalności programu poprzez zapisanie własnych algorytmów w postaci makropoleceń języka VBA. Makro NUM2TXT dokonuje konwersji liczby na jej tekstową reprezentację w języku polskim bądź angielskim. W artykule opisane zostały możliwości oferowane przez ten program wraz z przykładami użycia. Przedstawione zostały również podstawowe funkcje wchodzące w jego skład tak, aby ułatwić dopisanie nowych części odpowiadających za tłumaczenia na inne języki, poza polskim i angielskim.

* * *

1. Zastosowanie arkuszy kalkulacyjnych

Arkusze kalkulacyjne (patrz [11]) są jednymi z najbardziej użytecznych programów w codziennych zastosowaniach w małych i średnich firmach oraz zwykłych gospodarstwach domowych. Pozwalają nie tylko przechowywać dane, ale i dokonywać ich automatycznej obróbki oraz graficznej prezentacji wyników. Dzięki tym możliwościom stanowią nieocenione narzędzie w rękach księgowych i finansistów, lecz również szeregu badaczy nauk matematyczno-przyrodniczych, społeczno-ekonomicznych i humanistycznych. Arkusze kalkulacyjne udostępniają często dodatkowe narzędzia oferujące naprawdę zaawansowane funkcje. Przykładowo Microsoft Excel z pakietu Microsoft Office (zob. [2] lub [4]) zawiera

narzędzie Solver służące do rozwiązywania zadań optymalizacyjnych. Opis jego możliwości można znaleźć w [12]. Większość arkuszy pozwala również na rozbudowę oferowanych możliwości poprzez tworzenie własnych funkcji i podprogramów. Dzięki temu można lepiej dostosować program do własnych potrzeb, aby efektywniej wykorzystać jego potencjał.

Z pewnością jednym z najbardziej znanych arkuszy kalkulacyjnych jest Microsoft Excel. Niestety jest on programem komercyjnym, a w dodatku dostępnym tylko pod systemem operacyjnym Windows. Od roku 2000 coraz większą rzeszę zwolenników zdobywa **OpenOffice.org Calc** – zaawansowany arkusz kalkulacyjny wchodzący w skład bezpłatnego pakietu biurowego OpenOffice.org (patrz [3]), dostępny między innymi na platformach Microsoft Windows, Linux oraz Solaris. Funkcjonalnie odpowiada on programowi Microsoft Excel, a w dodatku obsługuje format dokumentów w nim zapisanych. Dzięki temu stanowi doskonałą alternatywę dla środowisk biurowych i uczelnianych posługujących się Microsoft Office.

1.1. Makra w VBA

Makro lub **makropolecenie** (porównaj [10]) to program przeznaczony do wykonywania przez określoną aplikację, na przykład edytor tekstu czy arkusz kalkulacyjny. Pisany jest zwykle w celu automatyzacji pracy z dokumentami. Jednym z najbardziej rozpowszechnionych języków programowania do tworzenia makropoleceń zaimplementowanych w wielu aplikacjach, między innymi Microsoft Office, AutoCAD (patrz [7]) czy WordPerfect (patrz [6]) jest *Visual Basic for Applications*, w skrócie VBA (patrz [1]). Stanowi on uproszczoną wersję języka Visual Basic (patrz [5]). Począwszy od wersji 2.0 OpenOffice.org również interpretuje makra zapisane w VBA.

Pod adresem www.oocomacros.org dostępny jest zbiór makr napisanych pod Open Office wraz z krótkimi dokumentacjami. Podzielone są one na dwie części. Pierwsza z nich przeznaczona jest dla użytkowników końcowych, którzy nie są zainteresowani pisaniem ani modyfikacjami makr, a tylko ich wykorzystaniem do własnych celów. Druga część natomiast przeznaczona jest dla osób chcących programować samodzielnie makra, rozbudowując w ten sposób standardowe możliwości pakietu Open Office. Najczęściej pisane są makra rozszerzające funkcjonalność edytora tekstu Open Office Writer oraz arkusza kalkulacyjnego Open Office Calc.

Dostępnych jest szereg pozycji książkowych do nauki języka VBA oraz pisania makr. Większość z nich opisuje makra pod kątem programu Microsoft Excel (patrz [13], [14], [15]). Obowiązkową pozycję dla osób rozpoczynających swoją przygodę z pisaniem makropoleczeń pod Open Office stanowi stale aktualizowany podręcznik napisany przez Andrew Pitonyak'a dostępny pod adresem www.pitonyak.org/oo.php [9]. W książce tej znajduje się nie tylko wprowadzenie do języka VBA lecz również szereg bardzo przydatnych przykładów gotowych do użycia makr. W celu zobrazowania, jak potężnym narzędziem w rękach wpraw nego programisty są makropolecenia, przytoczono zastosowania kilku przykładowych makr z tego podręcznika:

- otworzenie dokumentu z podanego adresu URL,
- wysłanie e-mailem dokumentu po uprzednim zabezpieczeniu go hasłem,
- wydrukowanie konkretnej strony w ustawionym formacie,
- wstawienie grafiki o ustalonym rozmiarze,
- rozpakowanie pliku lub katalogu spakowanego ZIPem,
- dokonanie konwersji danych zapisanych w kolumnach na format wierszowy,
- usunięcie z dokumentu pustych linii i zbędnych białych znaków (spacji, tabulatorów, itp.).

Możliwości są jednak o wiele większe. W VBA można nawet stworzyć makropolecenia tworzące graficzne obiekty służące do interakcji z użytkownikiem, włącznie z przechwytywaniem ruchów myszki i klawiszy wciskanych na klawiaturze. Poznanie podstawowych tylko możliwości oferowanych przez VBA może okazać się niezwykle pomocne w efektywniejszym wykorzystaniu używanych programów i dostosowaniu ich do własnych potrzeb.

1.2. Polskie wersje arkuszy kalkulacyjnych

Rodzime produkcje stanowią znikomy procent dostępnych na rynku aplikacji i są zwykle pisane na potrzeby środowisk akademickich. Praktyczne znaczenie mają obecnie tylko i wyłącznie rozbudowane pakiety biurowe takie jak wspomniane już Open Office czy Microsoft Office. Sytuacja ta nie powinna dziwić zważywszy na fakt, iż napisanie tak dużych programów wymaga wielu lat pracy co najmniej kilkudziesięcioosobowego sztabu ludzi. Nowo wprowadzane na rynek programy biurowe nie są przecież całkowicie nowymi aplikacjami, a jedynie ulepszonymi wersjami już istniejących produktów.

Polskie wersje językowe aplikacji powstają równolegle z oryginalnymi wersjami lub są udostępniane w późniejszych terminach niż premiera produktu

w oryginalnej wersji językowej. Jednak nawet te wersje nie są nowymi programami, a jedynie tłumaczeniami pierwowzorów. Za działanie każdego z tych programów odpowiadają bowiem te same algorytmy, a zmieniony jest jedynie interfejs użytkownika, który wyświetla komunikaty i instrukcje w konkretnym języku. Takie tłumaczenia nie mogą tym samym odzwierciedlić specyfiki danego języka. Szczególnie cierpi na tym język polski, który zawiera wiele konstrukcji nieobecnych w innych językach. Przykładowo tłumaczenie dat z języka angielskiego na większość innych języków można osiągnąć poprzez proste zastąpienie nazw miesięcy ich odpowiednikami w danym języku. Niestety w języku polskim nie daje to poprawnych rezultatów. Zamiast bowiem „1 kwietnia 2009” otrzymamy „1 kwiecień 2009”. Jest to jeden z najczęściej spotykanych przykładów, w których odmiana wyrazów nie została uwzględniona przy tłumaczeniu. Nie ma co liczyć w najbliższej przyszłości na włączenie do standardowych dystrybucji rozwiązania tego i innych problemów, jakie sprawia specyfika naszego ojczystego języka przy automatycznym tłumaczeniu. Najprostszym wyjściem okazuje się ściągnięcie odpowiedniego spolszczenia udostępnianego na ogół w postaci makropolecenia. Oczywiście nie wszystkie takie spolszczenia zostały napisane lub udostępnione dla ogółu użytkowników.

2. Problem konwersji liczb na słowne odpowiedniki

W wielu dokumentach związanych z finansami konieczne jest wypełnienie pola kwotą zapisaną słowami. Standardowo funkcja dokonująca tłumaczenia liczby na jej tekstowy odpowiednik nie jest udostępniona ani w arkuszu Microsoft Excel, ani w Open Office Calc. W omawianej tu książce [9] można znaleźć przykłady makropoleceń dokonujących podstawowych konwersji liczb na język angielski (rozdział *convert number to words*). Stosunkowo łatwo jest zaadaptować te przykłady do użycia w innym języku, na przykład niemieckim. Niestety działania przestają być trywialną sprawą w przypadku języka polskiego. Rozpatrzmy bowiem kilka prostych przykładów liczbowych¹⁰⁰ i ich zapis w notacji dziesiętnej:

- 2,01 – dwa i jedna setna,
- 101,02 – sto jeden i dwie setne,
- 211,12 – dwieście jedenaście i dwanaście setnych.

¹⁰⁰ Zgodnie z regułą używaną w większości współczesnych arkuszy kalkulacyjnych oraz w języku polskim do oddzielenia części całkowitej od części ułamkowej używamy przecinka, a nie kropki.

Zwróćmy uwagę, iż w zależności od części ułamkowej należy użyć słowa *setna*, *setne* lub *setnych*. Warto również zwrócić uwagę na użytą formę liczebników *jeden* oraz *dwa* w zależności od położenia cyfry w zapisie liczby. Przykładowo w pierwszym z przykładów należy użyć formy *jedna*, w drugim *jeden* (lub *sto*), natomiast w trzecim z przykładów *jedynka* pojawia się między innymi w postaci liczby *jedenaste*. Cyfra *dwa* pojawia się w formach *dwa*, *dwie*, *dwanaście* oraz *dwieście*.

Nie jest też łatwo określić reguły tworzenia bardziej złożonych liczb. Wystarczy spojrzeć na liczby z zakresu od *jedenastu* do *dziewiętnastu*. Czasami należy dodać do odpowiadającej cyfry końcówkę *-naście* (np. *dwa* – *dwanaście*) czasami *-aście* (np. *jeden* – *jedenaste*), czasami niektóre litery trzeba obciąć (np. *cztery* – *czternaście*), a czasami całkowicie zmodyfikować (np. *pięć* – *piętnaście*). Równie ciężko jest określić liczby *dziesiątek* poprzez dodanie odpowiednich końcówek *-dziesiąta*, *-dziesięci* czy *-dziesiąt*. Analogiczne sytuacje występują przy liczbach *setek*, *tysięcy*, *milionów*, itd. Problemy są również z odmianami słów *tysiąc* (*tysiące*, *tysięcy*), *milion* (*miliony*, *milionów*), itp. Okazuje się, że przewidzenie wszystkich możliwości występujących w języku polskim nie jest taką łatwą sprawą.

Format matematyczny nie jest przecież jedynym używanym. Częściej istnieje zapotrzebowanie na przedstawienie liczby w formacie walutowym. Spójrzmy na kilka przykładów:

- 123123123,12 zł – sto dwadzieścia trzy miliony sto dwadzieścia trzy tysiące sto dwadzieścia trzy złote i dwanaście groszy,
- 1000000000,1 \$ – miliard dolarów i dziesięć centów,
- 999,99 € – dziewięćset dziewięćdziesiąt dziewięć euro i dziewięćdziesiąt dziewięć eurocentów.

Na pierwszy rzut oka widać, że i tutaj mamy do czynienia z wieloma formami użytych wyrazów, na przykład *złoty*, *złote*, *złotych* czy *cent*, *centy*, *centów*. Warto też zwrócić uwagę na drugi z powyższych przykładów, aby uświadomić sobie, że nawet pomijając odmianę, istnieją znaczące różnice między językami. Tłumaczenie na język angielski (w wersji amerykańskiej) brzmiałoby *one billion dollars and ten cents*. Duże liczby mają zupełnie inne nazwy w języku angielskim niż w języku polskim, choć często dość podobnie brzmiące.

Tabela 1. Nazwy wybranych liczebników w języku polskim i angielskim

Liczba	Nazwa polska	Nazwa angielska
$10^3 = 1000$	tysiąc	thousand
$10^6 = 1000000$	milion	million
$10^9 = 1000000000$	miliard	billion
$10^{12} = 1000000000000$	bilion	trillion
$10^{15} = 1000...0000000$	biliard	quadrillion
$10^{18} = 1000...0000000$	trylion	quintillion
$10^{21} = 1000...0000000$	tryliard	sexillion
...	...	...
$10^{63} = 1000...0000000$	decyliard	vigintillion

Źródło: opracowanie własne

2.1. Makropolecenie NUM2TXT

Przedstawiony problem dokonania konwersji liczby na jej tekstową reprezentację może być oczywiście rozwiązany poprzez napisanie odpowiedniego makropolecenia. Kod źródłowy makropolecenia autorstwa M. Bernardelli i M. Kozakiewicza udostępniony został pod adresem:

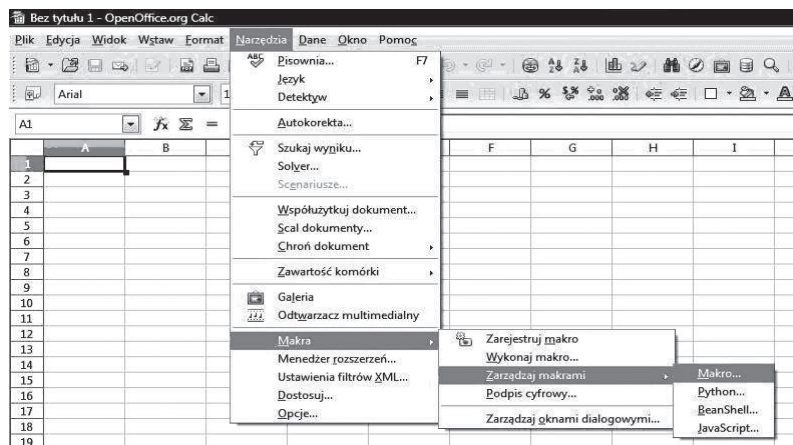
www.wsm-ciech.com

w zakładce *Rocznik Naukowy*.

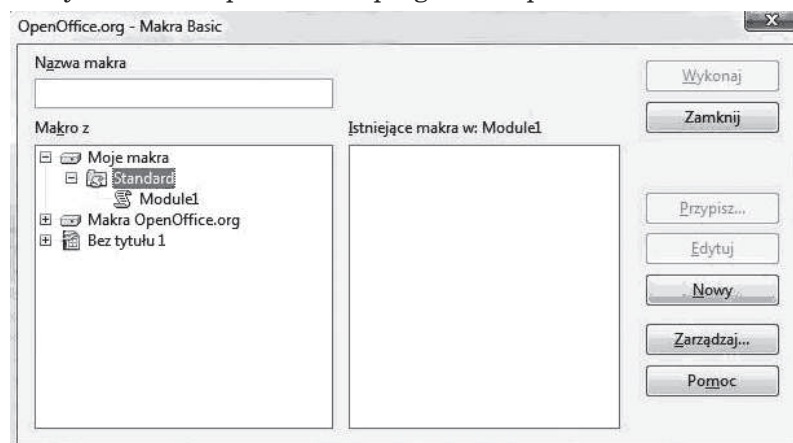
Po ściągnięciu ze strony należy makropolecenie zainstalować. W tym celu wybieramy z menu głównego programu Open Office Calc (wersja 3.0.1) zakładkę *Narzędzia* → *Makra* → *Zarządzaj makrami* → *Makro...* (patrz rys. 1)¹⁰¹.

¹⁰¹ Wszystkie rysunki są opracowaniem własnym autorów.

Algorytmy konwersji wartości liczbowych...



Rys. 1. Wczytanie makropolecenia w programie Open Office Calc



Rys. 2. Jedno z okien dialogowych podczas instalacji makropoleceń

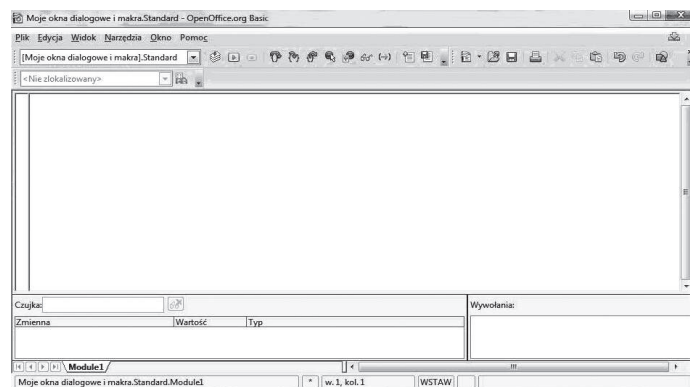
Pojawi się wówczas edytor (patrz rys. 3), w którego główne okno należy skopiować całą treść ściągniętego pliku *num2txt.txt*. Po zachowaniu zmian program Calc zostanie wzbogacony o nową funkcję *NUM2TXT*. Funkcja ta przyjmuje trzy argumenty

$$=NUM2TXT(number; language; format)$$

gdzie:

- *number* jest liczbą z zakresu od -10^{63} do 10^{63} .
- *language* jest parametrem opcjonalnym oznaczającym język w jakim powinna być przedstawiona tekstowa reprezentacja liczby *number*. Obsługiwane wartości parametru to:

- „PL” – język polski (domyślny),
 - „EN” – język angielski.
- *format* jest opcjonalnym parametrem określającym format w jakim powinna być przedstawiona tekstowa reprezentacja liczby number. Rozpoznawane wartości tego parametru to:
- „PL” – waluta polska (złoty),
 - „EU” – waluta europejska (euro),
 - „US” – waluta amerykańska (dolar),
 - „MAT” – notacja matematyczna (domyślna).



Rys. 3. Edytor makr w programie Open Office Calc

	A	B
1	=NUM2TXT(123,65)	sto dwadzieścia trzy i sześćdziesiąt pięć setnych
2	=NUM2TXT(123456789;"EN")	hundred twenty three million four hundred fifty six thousand seven hundred eighty nine
3	=NUM2TXT(987654321,01;"PL";"EU")	dziewięćset osiemdziesiąt siedem milionów sześćset pięćdziesiąt cztery tysiące trzysta dwadzieścia jeden euro i jeden eurocent
4	=NUM2TXT(100000000000000,99;"PL";"MAT")	jeden miliard jeden
5	=NUM2TXT(0,31;"EN";"PL")	thirty one groszes
6	=NUM2TXT(987,789;"EN";"US")	nine hundred eighty seven dollars and seventy nine cents
7		

Rys. 4. Przykłady użycia funkcji NUM2TXT

Warto poświęcić trochę miejsca na uwagi dotyczące działania funkcji NUM2TXT. Przede wszystkim wszystkie wprowadzane liczby są zaokrąglane do dwóch miejsc po przecinku, na przykład (patrz też wiersz szósty na rys. 4):

$=\text{NUM2TXT}(0,543)$

da w wyniku *pięćdziesiąt cztery setne*.

Makro obsługuje liczby aż do sześćdziesiątej trzeciej potęgi, ale ograniczeniem jest sposób przechowywania liczb w Calcu¹⁰², a mianowicie z dokładnością tylko do piętnastu cyfr znaczących. Oznacza to, że wprowadzając 18-cyfrową liczbę 123 456 789 123 456 789 dostajemy w istocie liczbę zaokrągloną do 15 cyfr znaczących: 123 456 789 123 457 000, przy czym ostatnie trzy cyfry zostaną zastąpione zerami. Dlatego mimo, iż makro obsługuje liczby większe niż 15-cyfrowe, to ze względu na zaokrąglenia dokonywane przez samą aplikację tak duże liczby nie będą poprawnie przetłumaczone. Przykładowo liczba $10^{18} + 1$, to jest:

=NUM2TXT(1000000000000000001)

zostanie przetłumaczona jako jeden trylion zamiast jeden trylion jeden. Na szczęście w praktyce raczej nikt nie operuje nazwami tak dużych liczb.

2.1. Kod źródłowy makra NUM2TXT

Makropolecenie *NUM2TXT* składa się z kilku funkcji i procedur. Główna z nich nosi właśnie nazwę *num2txt*. Poniżej znajduje się skrócony kod źródłowy tej funkcji. Osoby chcące rozbudować makro o obsługę innego, poza polskim i angielskim, języka muszą dopisać odpowiednie instrukcje w zaznaczonym wytłuszczeniu miejscu. Analogicznie należy postąpić w sytuacji dodawania innego formatu.

*Function num2txt(num AS Double, Optional lang AS String, Optional
format AS String) AS String*

DIM format_local AS Integer

If NOT (IsNumeric(num)) Then

num2txt = „Parameter: „ & CStr(num) & „ is not a number”

ElseIf ((num > PRECISION_HIGH) OR (num < PRECISION_LOW)) Then

num2txt = „Number: „ & CStr(num) & „ out of range”

Else

‘ parameter num is a number and is within a proper range

‘ check whether parameter format is present

¹⁰² Program Microsoft Excel ma dokładnie takie same ograniczenia.

```
If (IsMissing(format)) Then
    ' mat format
    format_local = 0
Else
    Select Case format
        Case „pl”, „PL”
            format_local = 1
        Case „eu”, „EU”
            format_local = 2
        Case „us”, „US”
            format_local = 3
        „,,,,,,,,,“ NOWY FORMAT „,,,,,,,,,“
        Case Else
            format_local = 0
    End Select
End If

    ' call the function translating the number to the language lang
Select Case lang
    Case „en”, „EN”
        num2txt = num2txt_en(num,format_local)
    „,,,,,,,,,“ NOWY JEZYK „,,,,,,,,,“
    Case Else
        num2txt = num2txt_pl(num,format_local)
End Select
End If
End Function ' num2txt
```

Funkcja *num2txt* na podstawie parametru *language* wywołuje odpowiednią funkcję odpowiadającą za konwersję na wybrany język. W przypadku języka angielskiego jest to funkcja *num2txt_en*, a dla języka polskiego jest to funkcja *num2txt_pl*. Poniżej znajduje się krótki opis funkcji, które zawiera część związana z tłumaczeniem na język polski. Aby dodać nowy język wystarczy na ich podstawie utworzyć analogiczne funkcje zwracające odpowiednie słowa z tego języka.

Function num2txt_pl(ByVal num AS Double,

ByVal format AS Integer) AS String

główna funkcja odpowiedzialna za konwersję liczby *num* na tekst w języku polskim i formacie *format*.

*Function translate1_pl(ByVal n AS Integer,
Optional ByVal format AS Integer) AS String*
funkcja zwracająca tekstowy odpowiednik w języku polskim
liczby naturalnej n mniejszej od 20

Function translate10_pl(ByVal n AS Integer) AS String
funkcja zwracająca tekstowy odpowiednik w języku polskim
liczby naturalnej n oznaczającej liczbę dziesiątek

Function translate100_pl(ByVal n AS Integer) AS String
funkcja zwracająca tekstowy odpowiednik w języku polskim
liczby naturalnej n oznaczającej liczbę setek

*Function translate_pl(ByVal n AS Integer,
Optional ByVal format AS Integer) AS String*
funkcja zwracająca tekstowy odpowiednik w języku polskim
liczby naturalnej n mniejszej od 1000

*Function translate_name_pl(ByVal n AS Integer,
ByVal k AS Integer) AS String*
funkcja zwracająca nazwę w języku polskim odpowiedniej potęgi
liczby dziesięć (patrz tabela 1), tj. tysiąc, milion, miliard, bilion, itd.

*Function name_before_pl(ByVal n AS Integer,
ByVal format AS Integer) AS String*
funkcja zwracająca słowo w języku polskim oznaczające jednostkę
walutową określoną parametrem format odpowiadającą całkowitej
części liczby, tj. złoty, euro, dolar

*Function name_after_pl(ByVal n AS Integer,
ByVal format AS Integer) AS String*
funkcja zwracająca słowo w języku polskim oznaczające jednostkę
walutową lub matematyczną określoną parametrem format
odpowiadającą ułamkowej części liczby, tj. grosz, eurocent, cent, setna

3. Podsumowanie

Komputery i aplikacje komputerowe mają za zadanie ułatwiać ludziom życie. Odkrywanie nowych zastosowań wiąże się ściśle z ich rosnącą rolą. Niemożliwym jest jednak przewidzieć przy tworzeniu programu komputerowego wszystkich możliwości zastosowań przy tak wielkiej liczbie ludzi. Dostosowanie programów do własnych potrzeb stanowi zatem kluczową sprawę przy automatyzacji przetwarzania danych. Najlepszym narzędziem do tego wydaje się tworzenie makro-

poleceń. Makro *NUM2TXT* jest tylko jednym, dość rozbudowanym co prawda, przykładem takiego rozszerzenia funkcjonalności programu Open Office Calc. Inne makra powstają każdego dnia i są często udostępniane przez autorów pozostałym użytkownikom. Zapotrzebowanie na nie pisze samo życie.

Źródła i bibliografia

1. <http://msdn.microsoft.com/en-us/isv/bb190538.aspx>
2. <http://office.microsoft.com/pl-pl/excel/default.aspx>
3. <http://openoffice.pl/index.php>
4. http://pl.wikipedia.org/wiki/Microsoft_Excel
5. http://pl.wikipedia.org/wiki/Visual_Basic
6. <http://pl.wikipedia.org/wiki/WordPerfect>
7. <http://www.autodesk.pl>
8. <http://www.oocomacros.org>
9. <http://www.pitonyak.org/oo.php>
10. <http://www.sjp.pl/co/makropolecenie>
11. <http://www.winter.pl/internet/arkusze.html>
12. Bernardelli M. (2008); Użycie Microsoft Excel do rozwiązywania zadań optymalizacyjnych, Rocznik Naukowy Wydziału Zarządzania w Ciechanowie, Tom II, Zeszyt 3–4, s. 77–90.
13. Lewandowski M. (2007); Tworzenie makr w VBA dla Excela 2003/2007. Ćwiczenia. Helion.
14. McFedries P. (2008); Microsoft Office 2007. Język VBA i makra. Rozwiązania w biznesie. Helion.
15. Walkenbach J. (2004); Excel 2003 PL. Programowanie w VBA. Vademecum profesjonalisty. Helion.