1. Zmienne. Typy.

Dużą zaletą Visual Basica jest jego modułowość. Cały program jest budowany ze ściśle od siebie odseparowanych składowych - procedur. Procedura pobiera ze swojego środowiska (od użytkownika, z innych procedur) informacje, przetwarza je w określony algorytmem sposób i wyprowadza je na zewnątrz. Informacje te są przechowywane jako wartości cech (właściwości) kontrolek (elementów sterujących) oraz zmiennych.

Każdą zmienną określa sześć parametrów:

1. Nazwa zmiennej - używana jest w kodzie aby odwołać się do zmiennej
2. Adres - miejsce w pamięci, gdzie przechowywana jest wartość zmiennej
3. Typ danych - typ i rozmiar początkowych danych, które zmienna może przyjmować
4. Wartość - umieszczona jest w pamięci pod adresem zmiennej
5. Zasięg - blok kodu, który ma dostęp do zmiennej
6. Czas życia - czas, w którym zmienna istnieje w pamięci i w którym kod może się do niej odwoła

W Visual Basicu nie ma obowiązku deklarowania zmiennych prostych. Jest to jednak zalecane. Standardowym typem danych jest typ wariantowy. W celu określenia specyficznego typu zmiennej należy użyć instrukcji Dim. Instrukcje tworzą drugą składową procedur tworzących aplikację. Określają one w jaki sposób mają być przetwarzane wartości zmiennych.

Dostępne są dwa rodzaje typów zmiennych:

- proste, których budowa jest zdefiniowana w samym języku programowania
- złożone, których budowę określa programista, komponując je ze zmiennych prostych.

Proste typy danych to :

1. Integer - 2- bajtowa liczba całkowita
2. Long - 4- bajtowa liczba całkowita
3. Single - 4- bajtowa liczba zmiennoprzecinkowa
4. Double - 8- bajtowa liczba zmiennoprzecinkowa
5. Currency - 8- bajtowa liczba stałoprzecinkowa

6. String - łańcuch znaków (do 64kB)
7. Variant - Data / czas / liczba jak Double
   / łańcuch jak String

Deklaracja zmiennej ma postać

Dim nazwa_zmiennej **As** typ_danych

Jeżeli chce się przechowywać wartości z poprzednich wywołań procedury, zamiast słowa **Dim** należy użyć **Static.**

Zakres zmiennej określa dostęp do zmiennej przez różne części programu.

Wyróżniamy:

- Zmienne lokalne - dostępne tylko w procedurze, w której zostały zdefiniowane.
- Zmienne modułowe - dostępne dla wszystkich procedur w danym module.
- Zmienne ogólne - dostępne dla wszystkich procedur programu

2. Podstawowe instrukcje.

Instrukcja podstawienia, zmiennej *Zmienna* przypisuje wartość *wyrażenie*

Zmienna = wyrażenie

Instrukcja warunkowa.

- Jeżeli wyrażenie jest prawdą wykonywana jest instrukcja.

If wyrażenie **Then** instrukcja

- Jeżeli prawdziwe jest wyrażenie wykonywane są *Instrukcje1*, w przeciwnym wypadku wykonywane są *Instrukcje2***.**

If wyrażenie **Then**
Instrukcje1
Else
Instrukcje2
End If

Pętla **For**, instrukcje wykonywane są do momentu, gdy zmienna osiągnie wartość końcową.

For zmienna = wartośćPoczątkowa **To** wartośćKońcowa
instrukcje
Next zmienna

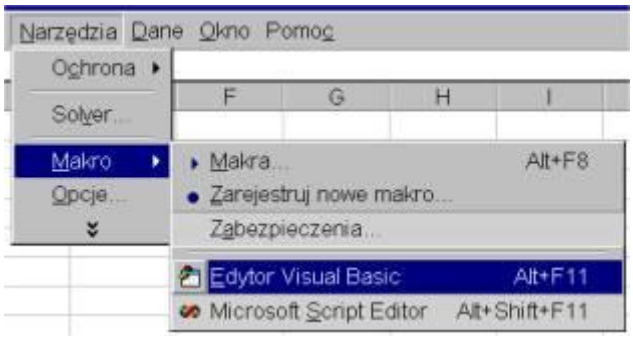
lub

For zmienna = wartośćP **To** wartośćK **Step** krok
instrukcje
Next zmienna

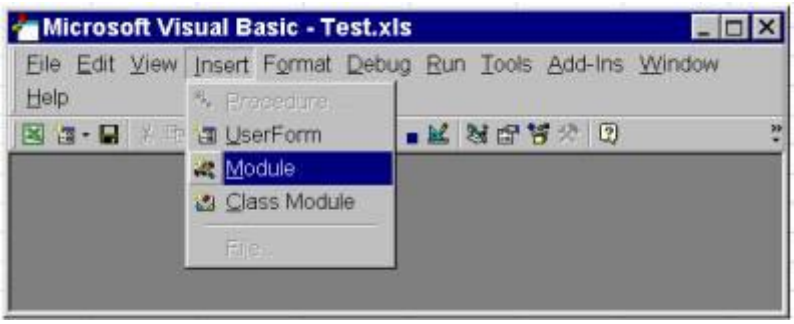
w tym przypadku **Step** oznacza o ile ma się zmieniać wartość zmiennej między kolejnymi iteracjami.

3. Własne funkcje w Excelu.

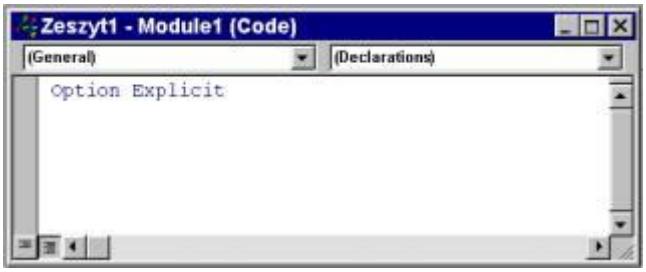
Aby pisać własne należy w Excelu z menu *Narzędzia* wybrać opcję *Makra* i dalej *Edytor Visual Basic*.



otworzymy w ten sposób następujące okno:



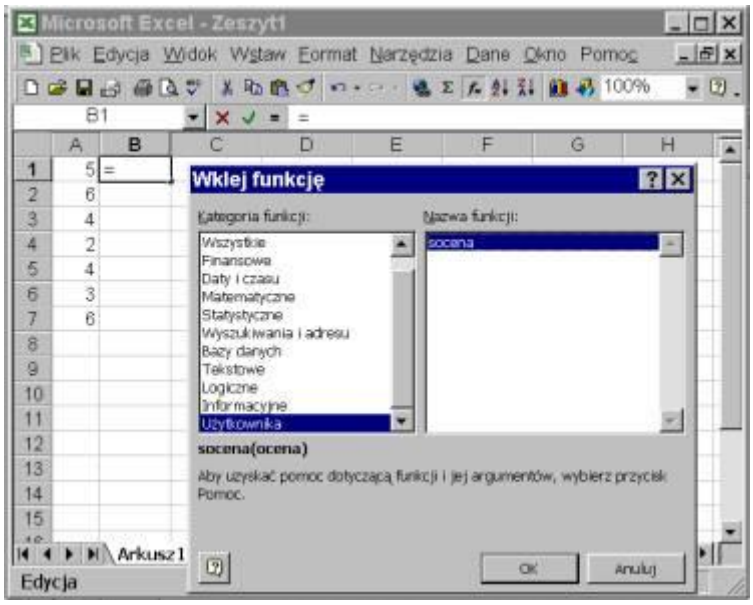
w którym po wybraniu opcji *Module z* menu *Insert* zobaczymy okno umożliwiające wpisanie kodu funkcji.



Dla przykładu wpiszmy kod funkcji, która ocenę zapisaną cyfrowo zamienia na zapis słowny.

```
Function socena(ocena As Integer) As String
    Dim napis As String 'deklaracja zmiennej
    napis = ""    'zmiennej napis przypisano łańcuch pusty
    If ocena = 6 Then napis = "celujący" 'jeśli ocena = 6, to
                      'za zmienną napis podstaw celujący
    If ocena = 5 Then napis = "bardzo dobry"
    If ocena = 4 Then napis = "dobry"
    If ocena = 3 Then napis = "dostateczny"
    If ocena = 2 Then napis = "dopuszczający"
    If ocena = 1 Then napis = "niedostateczny"
    socena = napis      'nazwie funkcji przypisz wartość
                        'zmiennej napis
End Function
```

Po wpisaniu powyższego kodu wracamy do okna Excela, wpisanej funkcji możemy użyć klikając w ikonę *Wklej funkcję* standardowego paska narzędzi. Otworzymy okno:



W kategorii funkcji *Użytkownika* wybierzmy zdefiniowaną nazwę funkcji *socena* i naciśnijmy *OK*. Zaznaczmy komórkę zawierającą ocenę (lub wpiszmy odpowiedni adres komórki) i naciśnijmy *Enter*. Kopiując formułę do odpowiedniego zakresu otrzymamy słowny zapis ocen.

Uwaga.

Sposób przekazywania argumentów określony jest w deklaracji funkcji. Argumenty mogą być przekazywane poprzez referencję (domyślnie), lub poprzez wartość (nazwę argumentu w deklaracji funkcji należy poprzedzić słowem **ByVal**). Jeśli argumenty przekazywane są poprzez wartość, to pobierana jest kopia danych przekazywanych jako argumenty wejściowe.

4. Funkcje standardowe.

W kolejnym przykładzie wykorzystamy kilka standardowych funkcji działających na tekstac

Len (*napis*) - zwraca liczbę znaków występujących w napisie.

Mid(*napis*, *pozycjaPoczątkowa*, *liczbaZnaków*) - wycina z napisu *napis* łańcuch o długości określonej wyrażeniem *liczbaZnaków*,

począwszy od znaku o numerze *pozycjaPoczątkowa*.

Left(*napis*, *liczbaZnaków*) - wycina z napisu *napis* określoną wyrażeniem *liczbaZnaków* liczbę znaków z lewej

Wykorzystamy również operator **&**sklejania napisów.

Poniższa funkcja zamienia cyfry na napisy.

```
Function scyfra(cyfra As Integer) As String
    Dim napis As String
    scyfra = ""
    If cyfra = 0 Then napis = "zero"
    If cyfra = 1 Then napis = "jeden"
    If cyfra = 2 Then napis = "dwa"
    If cyfra = 3 Then napis = "trzy"
    If cyfra = 4 Then napis = "cztery"
    If cyfra = 5 Then napis = "pięć"
    If cyfra = 6 Then napis = "sześć"
    If cyfra = 7 Then napis = "siedem"
    If cyfra = 8 Then napis = "osiem"
    If cyfra = 9 Then napis = "dziewięć"
    scyfra = napis
End Function
```

W kodzie jednej funkcji można umieścić odwołania do innych. Ilustruje to poniższy przykład. Ta funkcja zamienia liczbę na zapis słowny jej cyfr, na przykład wywołana dla liczby 123 zwróci jeden-dwa-trzy.

```
Function sliczba(liczba) As String
    Dim i As Integer            'deklaracja zmiennej i typu
                                'liczba całkowita
    Dim napis As String         'deklaracja zmiennej typu
                                'łańcuchowego (napis)
    If Len(liczba) = 0 Then     'jeśli długość zapisu zmiennej
                                'liczba = 0, to
        napis = ""              'przypisanie łańcucha pustego
    Else
        napis = scyfra(Left(liczba, 1))
      'za napis podstawiamy zapis słowny pierwszej cyfry
      'realizujemy to przez wywołanie poprzedniej funkcji
        For i = 2 To Len(liczba)
            'dla kolejnych cyfr doklejamy do napisu
            napis = napis & "-" & scyfra(Mid(liczba, i, 1))
        Next i
    End If
    sliczba = napis
End Function
```

Dalsze standardowe funkcje działające na napisach (zmiennych typu **String**)

Right(*napis*, *liczbaZnaków*) - wycina z napisu *napis* określoną wyrażeniem *liczbaZnaków* liczbę znaków z prawej strony, np. **Right**("napis",2)="is"

UCase(*napis*) - zamienia małe litery występujące w napisie na wielkie, a pozostałe znaki pozostawia bez zmian

LCase(*napis*) - zamienia wielkie litery występujące w napisie na małe, a pozostałe znaki pozostawia bez zmian

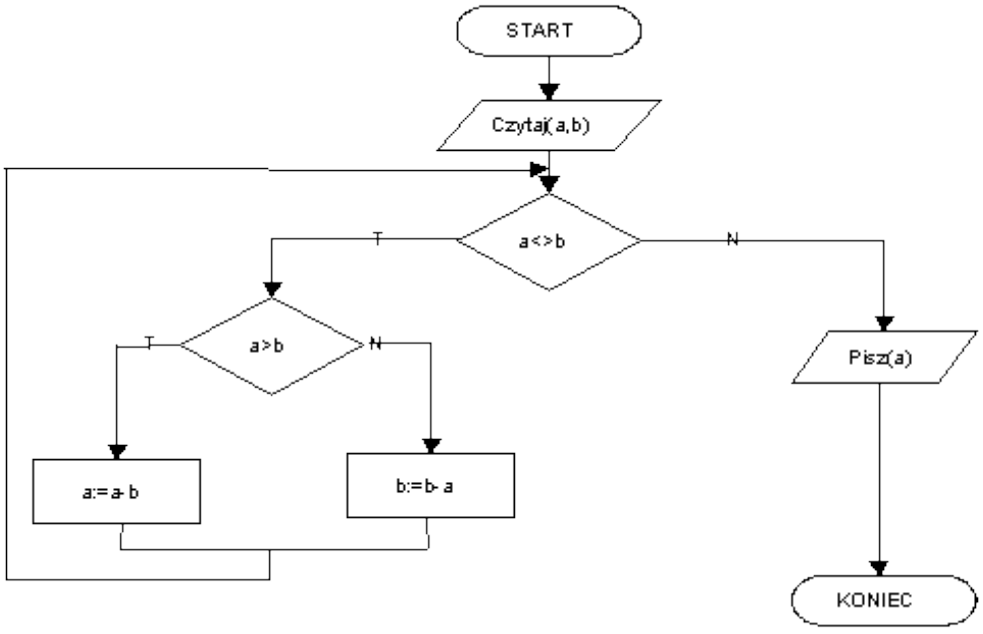
Zadania

1. Napisz funkcję zamieniającą zapis cyfrowy dnia tygodnia na zapis słowny.
2. Napisz funkcję zamieniającą zapis cyfrowy miesiąca na zapis słowny.
3. Napisz funkcję, której argumentem jest numer miesiąca a wynikiem kwartał, do którego dany miesiąc należy.
4. Napisz funkcję zamieniającą liczbę punktów na ocenę.

Algorytmy. Pętle Do ... Loop.

Informatyka jest dziedziną wiedzy i działalności zajmującą się między innymi algorytmami. **Algorytm** to zbiór reguł rozwiązania danego zadania w skończonej licznie kroków. Algorytm składa się z dwóch zasadniczych części: **opisu danych,** na których działa oraz **opisu czynności**, które są wykonywane na tych danych. Algorytm zapisany w języku programowania nazywa się **programem**, w którym czynności wykonywane na danych są opisywane za pomocą odpowiednich **instrukcji** języka. Żeby przekazać posiadany algorytm drugiemu człowiekowi lub maszynie musimy go zapisać. Jednym ze sposobów, w którym zapisuje się algorytmy w komunikacji człowiek - człowiek, jest język **schematów blokowych**.

Przykład 1. *Schemat blokowy algorytmu Euklidesa, wyznaczającego największy wspólny dzielnik dwóch liczb naturalnych większych od zera.*



Kolejne ważne instrukcje to pętla:

Do While warunek

instrukcje

Loop

w której instrukcję wykonywane są dopóki warunek jest prawdziwy,

oraz

Do

instrukcje

Loop Until warunek

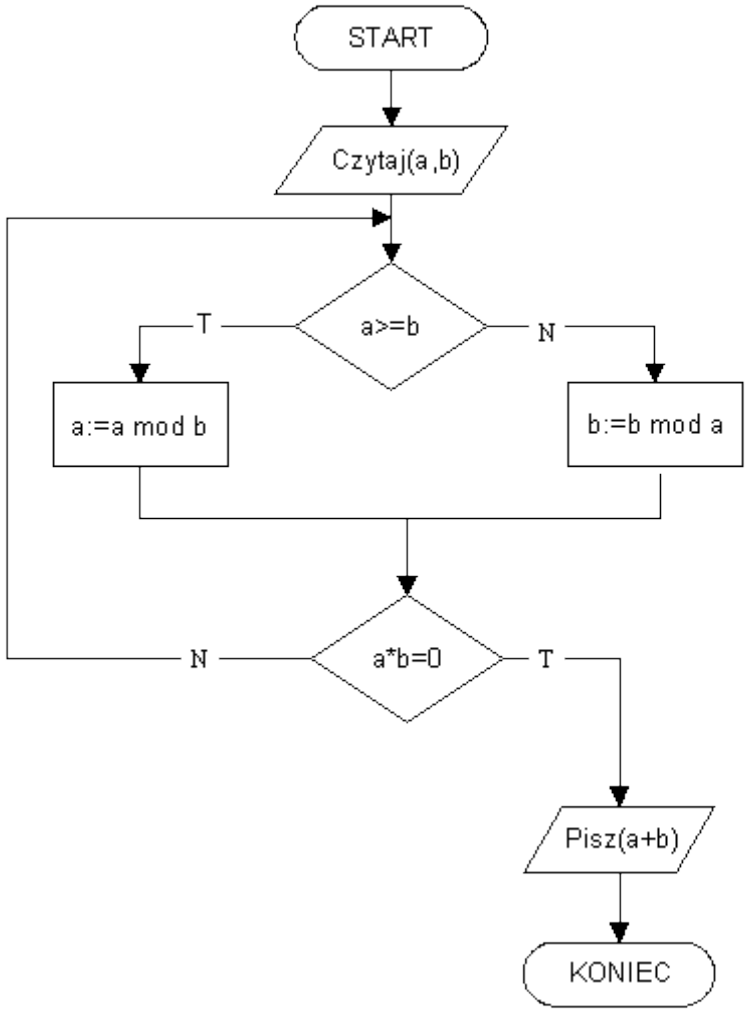
w której instrukcję wykonywane są do momentu, w którym warunek stanie się prawdziwy.

Kod funkcji realizującej algorytm Euklidesa ma postać:

```
Function nwd(a, b As Long) As Long 'zmienne a i b oraz wynik
                  ' funkcji są liczbami całkowitymi (długimi)
    Do While a <> b      'wykonuj dopóki a różne od b
        If a > b Then    'jeśli a większe od b, wtedy
            a = a - b    'za a podstaw różnicę a-b
        Else             'w przeciwnym przypadku
            b = b - a    'za b podstaw różnicę b-a
        End If           'koniec instrukcji warunkowej
    Loop                 'koniec pętli
    nwd = a              'nazwie funkcji przypisz wartość
                         'zmiennej a
End Function         'koniec funkcji
```

Istotnym zagadnieniem jest **złożoność algorytmiczna**. Obejmuje ona **złożoność czasową** mierzoną ilością, zależną od danych, podstawowych operacji, które trzeba wykonać w czasie realizacji algorytmu oraz **złożoność pamięciową** mierzoną ilością i wielkością użytych zmiennych, czyli ilością pamięci komputera.

Przykład 1a. *Schemat blokowy algorytmu, wyznaczającego największy wspólny dzielnik dwóch liczb naturalnych większych od zera o mniejszej złożoności czasowej niż algorytm Euklidesa*



Uwaga:

a **Mod** b oznacza resztę z dzielenia liczby a przez b.

Kod funkcji obliczającej NWD tym sposobem ma postać:

```
Function nwd2(a, b As Long) As Long
    Do                      'wykonuj
        If a >= b Then      'jeśli a jest większe lub równe b, to
            a = a Mod b     'za zmienną a podstaw resztę
                            'z dzielenia a przez b
        Else                'w przeciwnym przypadku
            b = b Mod a     'za zmienną b podstaw resztę
                            'z dzielenia b przez a
        End If              'koniec instrukcji warunkowej
    Loop Until a * b = 0        'sprawdzanie warunku na końcu
                                'pętli - wyjście z pętli, gdy
                                'iloczyn a i b równa się zero
    nwd2 = a + b                'przypisanie nazwie funkcji
                                'wartości sumy a i b
End Function
```

Zapis słowny liczb

```
Function J_slownie(ByVal CYFRA As Long) As String
    If CYFRA = 1 Then J_slownie = "jeden"
    If CYFRA = 2 Then J_slownie = "dwa"
    If CYFRA = 3 Then J_slownie = "trzy"
    If CYFRA = 4 Then J_slownie = "cztery"
    If CYFRA = 5 Then J_slownie = "pięć"
    If CYFRA = 6 Then J_slownie = "sześć"
    If CYFRA = 7 Then J_slownie = "siedem"
    If CYFRA = 8 Then J_slownie = "osiem"
    If CYFRA = 9 Then J_slownie = "dziewięć"
    If CYFRA = 10 Then J_slownie = "dziesięć"
    If CYFRA = 11 Then J_slownie = "jedenaście"
    If CYFRA = 12 Then J_slownie = "dwanaście"
    If CYFRA = 13 Then J_slownie = "trzynaście"
    If CYFRA = 14 Then J_slownie = "czternaście"
    If CYFRA = 15 Then J_slownie = "piętnaście"
    If CYFRA = 16 Then J_slownie = "szesnaście"
    If CYFRA = 17 Then J_slownie = "siedemnaście"
    If CYFRA = 18 Then J_slownie = "osiemnaście"
    If CYFRA = 19 Then J_slownie = "dziewiętnaście"
    If CYFRA > 19 Then J_slownie = ""
End Function

Function D_slownie(ByVal d As Long) As String
    If d = 2 Then
        D_slownie = J_slownie(d) & "dzieścia" & " "
    ElseIf d = 3 Then
        D_slownie = J_slownie(d) & "dzieści" & " "
    ElseIf d = 4 Then
        D_slownie = "czterdzieści" & " "
    ElseIf d = 5 Or d = 6 Or d = 7 Or d = 8 Or d = 9 Then
        D_slownie = J_slownie(d) & "dziesiąt" & " "
    Else
        D_slownie = ""
    End If
End Function



Function S_slownie(ByVal s As Long) As String
    If s = 1 Then
        S_slownie = "sto" & " "
    ElseIf s = 2 Then
        S_slownie = "dwieście" & " "
    ElseIf s = 3 Or s = 4 Then
        S_slownie = J_slownie(s) & "sta" & " "
    ElseIf s = 5 Or s = 6 Or s = 7 Or s = 8 Or s = 9 Then
        S_slownie = J_slownie(s) & "set" & " "
    Else
        S_slownie = ""
    End If
End Function
```

```
Function T_slownie(ByVal t As Long) As String
    Dim tt, st, dt, jt As Long
    tt = t
    st = tt \ 100
    tt = tt Mod 100
    dt = tt \ 10
    jt = t Mod 10
    If t = 1 Then
        T_slownie = J_slownie(t) & " tysiąc" & " "
    ElseIf t = 2 Or t = 3 Or t = 4 Then
        T_slownie = J_slownie(t) & " tysiące" & " "
    ElseIf (t >= 5) And (t <= 19) Then
        T_slownie = J_slownie(t) & " tysięcy" & " "
    ElseIf (t >= 20) And (t <= 99) And ((jt = 2) _
                        Or (jt = 3) Or (jt = 4)) Then
        T_slownie = D_slownie(dt) & J_slownie(jt) & _
        " tysiące" & " "
    ElseIf (t >= 20) And (t <= 99) And ((jt <> 2) _
                        And (jt <> 3) And (jt <> 4)) Then
        T_slownie = D_slownie(dt) & J_slownie(jt) _
        & " tysięcy" & " "
    ElseIf (t > 99) And ((jt = 2) Or (jt = 3) _
                        Or (jt = 4)) Then
        T_slownie = S_slownie(st) & D_slownie(dt) _
        & J_slownie(jt) & " tysiące" & " "
    ElseIf (t > 99) And ((jt <> 2) And (jt <> 3) _
                                    And (jt <> 4)) Then
        T_slownie = S_slownie(st) & D_slownie(dt) _
        & J_slownie(jt) & " tysięcy" & " "
    Else
        T_slownie = ""
 End If
End Function
```

```
Function M_slownie(ByVal m As Long) As String
    Dim mm, sm, dm, jm As Long
    mm = m
    sm = mm \ 100
    mm = mm Mod 100
    dm = mm \ 10
    jm = mm Mod 10
    If m = 1 Then
        M_slownie = J_slownie(m) & " milion" & " "
    ElseIf m = 2 Or m = 3 Or m = 4 Then
        M_slownie = J_slownie(m) & " miliony" & " "
    ElseIf (m >= 5) And (m <= 19) Then
        M_slownie = J_slownie(m) & " milionów" & " "
    ElseIf (m >= 20) And (m <= 99) And ((jm = 2) _
                        Or (jm = 3) Or (jm = 4)) Then
        M_slownie = D_slownie(dm) & J_slownie(jm) _
        & " miliony" & " "
    ElseIf (m >= 20) And (m <= 99) And ((jm <> 2) _
                        And (jm <> 3) And (jm <> 4)) Then
        M_slownie = D_slownie(dm) & J_slownie(jm) _
        & " milionów" & " "
    ElseIf (m > 99) And ((jm = 2) Or (jm = 3) _
                                    Or (jm = 4)) Then
        M_slownie = S_slownie(sm) & D_slownie(dm) _
        & J_slownie(jm) & " miliony" & " "
    ElseIf (m > 99) And ((jm <> 2) And (jm <> 3) _
                                    And (jm <> 4)) Then
        M_slownie = S_slownie(sm) & D_slownie(dm) _
        & J_slownie(jm) & " milionów" & " "
    Else
        M_slownie = ""
    End If
End Function
```

```
Function liczbaSlownie(liczba As Long) As String
    Dim slownie As String
    Dim J, d, s, t, m, tm As Long
    slownie = ""
    tm = liczba \ 1000000000
    If tm = 1 Then slownie = slownie & "miliard "
    If tm = 2 Then slownie = slownie & " dwa miliardy "
    liczba = liczba Mod 1000000000
    m = liczba \ 1000000
    liczba = liczba Mod 1000000
    slownie = slownie & M_slownie(m) & " "
    t = liczba \ 1000
    liczba = liczba Mod 1000
    slownie = slownie & T_slownie(t) & " "
    s = liczba \ 100
    slownie = slownie & S_slownie(s)
    liczba = liczba Mod 100
    If liczba < 20 Then
        slownie = slownie & J_slownie(liczba)
    Else
        d = (liczba \ 10)
        slownie = slownie & D_slownie(d)
        J = liczba Mod 10
        If J < 10 Then
            slownie = slownie & J_slownie(J)
        End If
    End If
    liczbaSlownie = slownie
End Function
```

```
Function BezSpacji(napis As String)
    Dim i As Integer
    Dim znak, NowyNapis
    NowyNapis = ""
    For i = 1 To Len(napis)
        znak = Mid(napis, i, 1)
    If znak <> " " Then
        NowyNapis = NowyNapis & znak
    End If
    Next i
    BezSpacji = NowyNapis
End Function
```

Przykładowe wywołanie funkcji LiczbaSlownie(2123456789) ma postać

dwa miliardy sto dwadzieścia trzy miliony
czterysta pięćdziesiąt sześć tysięcy
siedemset osiemdziesiąt dziewięć