Elementarz programowania
1. QBASIC

Cel ogólny: Podstawowe zasady programowania w QBASIC

Część 1
Elementy języka, tryby pracy, typy zmiennych i ich deklarowanie, wprowadzanie danych do programu, wyprowadzanie wyników

Spis treści

STRUKTURA JEZYKA PROGRAMOWANIA

- Każdy język programowania ma swoją
  - strukturę,
  - reguły składni
  - elementy języka

Elementy języka programowania

Język programowania składa się z:

- **słow kluczowych**, np. PRINT
- **deklaracji**, np. DIM I, J AS INTEGER
- **operatorów** , np. =, +, *, /, <, <>, NOT, AND
- **funkcji i podprogramów**, np. SQR, SIN

Słowa kluczowe

- W języku **Qbasic** używanych jest ponad 100 określonych słów (pochodzących z języka angielskiego) zwanych **słowami kluczowymi**. Są to części instrukcji języka, a także nazwy funkcji standardowych i niektórych operatorów (np. logicznych).

Pojęcia

- Słowa kluczowe i funkcje, uzupełnione ***parametrami i argumentami,*** mogą być użyte do tworzenia instrukcji, czyli poleceń dla komputera.

  Deklaracje są czymś w rodzaju informacji dla komputera o tym, co mają reprezentować wyszczególnione w nich *nazwy i wielkości.*

  Operatory to znaki reprezentujące ***działania arytmetyczne i logiczne.***

Instrukcje – polecenia dla komputera

- Słowa kluczowe i funkcje,
  uzupełnione ***parametrami i argumentami,***
  mogą być użyte do tworzenia instrukcji,
  czyli poleceń dla komputera.
- Przykłady instrukcji:
  IF A<>C THEN PRINT A-C

  INPUT "Jak masz na imie"; imie$

Stałe, zmienne

- **STAŁA** zwana czasem **literałem** – **przedstawia konkretną wartość** w sposób jawny, dosłowny czyli literalny,
  np. 10.0, Kowalski
- **ZMIENNA** to **symboliczne przedstawienie struktury danych określonego typu przy pomocy nazwy, np.**
  Nazwisko, alfa, dana2, wektor1(k)

Wprowadzenie do programu QBASIC

- **QBasic** jest językiem programowania wysokiego poziomu
- **Qbasic** jest programem interpretacyjnym. Program jest zintegrowany z edytorem tekstu
- QBasic to produkt firmy Microsoft. Zastąpił on **GW BASIC** dystrybuowany wraz z systemem operacyjnym MS-DOS

Uruchomienie programu:

- **QBASIC** [Enter] lub **QBASCIC** *Nazwa_pliku*
  Naciskamy **Esc** i można wpisywać program.
  Zapis programu przez **Alt F S** lub **Alt F A** (sAve As).
  Wyjście do menu górnego przez **Alt** lub Alt i litera menu górnego.
  Opuszczenie QBasic przez opcje **eXit** z menu **File**.
  Program najlepiej wpisywać małymi literami.
  Jeśli jest to słowo kluczowe to po naciśnięciu Enter, słowo zostanie zamienione na duże litery.
  Pozostałe słowa będą wpisane małymi literami.

Skróty klawiszowe w QBasic

- **Shift F1** - help
  F6 - tryb bezpośredni / tryb programowy - przełączanie
  Shift F5 - uruchomienie programu
  F1 na słowie programu - pomoc do słowa
  F4 - ekran wynikowy

  Uwaga!
  Apostrof ‘ lub słowo kluczowe REM
  umieszczone w wierszu programu powodują, ze cały tekst na prawo od niego jest komentarzem i nie ma wpływu na przebieg wykonania programu.

Tryby pracy w programie QBasic

- Po wywołaniu programu QBasic mamy 2 możliwości korzystania z niego, zwane trybami pracy:
  - **Tryb programowy**
  - **Tryb natychmiastowy**
- W trybie programowym kursor znajduje sie w górnym, dużym oknie edytora.
  Wpisywane instrukcje są umieszczone w pamięci komputera, a ich wykonanie następuje po naciśnięciu **Shift+F5**.
- Przejście do trybu natychmiastowego następuje w wyniku naciśnięcia klawisza **F6**.
  Kursor przeskakuje do małego okna na dole ekranu, zatytułowanego Immediate.
  W trybie tym zakończenie instrukcji wciśnięciem Enter jest równoznaczne poleceniem wykonania instrukcji.

Tryb **natychmiastowy**

- Tryb natychmiastowy pozwala korzystać z programu jak z **kalkulatora**.
  Przykład
  Naciskamy **F6** i przechodzimy do trybu **natychmiastowego**.
  Piszemy:
  PRINT 3.14*7^2 lub ***? 3.14*7^2*** i <**Enter**>
  Znika dotychczasowy obraz i pojawia sie **ekran wynikowy** i ostatnia widoczna na nim wielkość - 153.86 - jest wynikiem wykonania instrukcji wpisanej w trybie natychmiastowym.
  Wciśniecie dowolnego klawisza przywraca poprzednia postać ekranu, z oknami do redagowania programu i wpisywania instrukcji do natychmiastowego wykonania.

Tryby Qbasic, ekran wynikowy

- Powrót do trybu programowego przez **F6.**
  Tryb natychmiastowy jest pomocny podczas uruchamiania programów.
  W trybie tym są dostępne wszystkie zmienne używane w programie.
  Jeżeli w trakcie wykonywania programu wystąpił błąd i nastąpiło zatrzymanie programu, to możemy przejść do trybu natychmiastowego i wyświetlić wartości dowolnych zmiennych.
  QBasic pozwala na "przywołanie" **ekranu wynikowego** w dowolnym momencie redagowania programu. Realizuje to naciśniecie klawisza **F4.**

Przykłady z trybami, zadania

I. Przejdź do trybu natychmiastowego w Qbasicu – **F6** i wykonaj obliczenia:

1) pi=4*atn(1): ? Pi

2) R=2: ? Pi*r^2

3) a=3: b=2: ?a/b, a*b, a^b

II. Przejdź do trybu programowego F6. Napisz **CLS**. Przepisz powyższe instrukcje i uruchom Shift F5 (Run Start). Sprawdź klawisz F4

III. Przejdź do HELP i poszukaj opisu CLS. Skopiuj opis. Wklej do okna edytora. Zapisz plik. Poszukaj innych funkcji.

Deklaracje typów zmiennych

- W językach programowania rozróżnia sie 3 podstawowe typy **zmiennych** i **stałych**:
  - **INTEGER** - zmienne i stale liczbowe całkowite
  - **REAL** - zmienne i stale liczbowe rzeczywiste
  - **STRING** - zmienne i stale znakowe
- Dla każdej zmiennej, użytej w programie, musi być zarezerwowana w komputerze pewna liczba komórek pamięci, zależna od typu zmiennej.
  Te rezerwacje pamięci zapewniają deklaracje, umieszczone na początku programu.
  Deklaracja jest informacją, ze w programie użyte będą zmienne o wymienionych nazwach i typach wynikających z rodzaju deklaracji.
  Obok zmiennych, w programie występują **stałe**, które też mogą być typu Integer, Real i String.

Zmienna, stała

- **Zmienna** to *nazwa, do której może być przypisana wartość.*
  Ta wartość nazywa sie stałą.
  W instrukcji
  liczbauczniow = 342
  liczbauczniow (nie może być polska litera) to ***zmienna***, a 342 to ***stala***, czyli konkretna liczba.
  Pokazana instrukcja przypisuje stałą 342 do zmiennej liczbauczniow.
- *Stała nie musi być przypisana do zmiennej.*
  Np. w instrukcji
  PRINT 8 / 3
  występuje słowo kluczowe PRINT i 2 stale: 8 i 3.

Typy zmiennych i stałych w QBasic

Nazwa typu	Określa	Zakres wartości	
		od	do
INTEGER	liczby całkowite	**-32 768**	**32 767**
LONG	liczby **całkowite długie**	**-2 147 483 648**	**2 147 483 647**
SINGLE	liczby rzeczywiste **pojedynczej** precyzji	**± 2.802597E-45**	**± 3.402823E+38**
DOUBLE	liczby rzeczywiste **podwójnej** precyzji	**±4.940656458412465D-324**	**±1.79769313486231D+308**
STRING	łańcuchy znaków czyli teksty	pusty	**32767** znaków

Zakres zmiennych QBASICa z Help

Znajdź zawartość tej pomocy i skopiuj do Painta

Help, Qbasic Evironment Limits, Enter

Item		Maximum	Minimum
Variable name length		40 characters	1 character
String length		32,767 characters	0 characters
Integers		32,767	-32,768
Long Integers		2,147,483,647	-2,147,483,648
Single precision numbers (positive)		3.402823E+38	2.802597E-45
Single precision numbers (negative)		-2.802597E-45	-3.402823E+38
Double precision numbers (positive)			
	Maximum:	1.797693134862310+308	
	Minimum:	4.940656458412465D-324	
Double precision (negative)			
	Maximum:	-4.940656458412465D-324	
	Minimum:	-1.797693134862310+308	

Deklarowanie zmiennych:

- W BASICu nie jest konieczne deklarowanie **typu** użytych zmiennych liczbowych i wtedy domyślnie są one takiego typu liczbowego jaki jest najczęściej stosowany:
  liczby rzeczywiste pojedynczej precyzji - SINGLE
  a tam gdzie będą wymagane liczby całkowite zostanie automatycznie pominięta część ułamkowa.
- Jednak na wyższym stopniu zaawansowania powinno się przestrzegać zasady deklarowania typów obowiązującej w innych językach albo przynajmniej być świadomym jakiego typu są użyte w programie zmienne.

Sposoby deklarowania zmiennych w QBasicu

I sposób:

W nowszych wersjach Basica zalecana deklaracja metodą:

przez deklaracje zapisane na początku programu lub jego podprogramu, mające postać:

DIM *lista nazw zmiennych* **AS typ**

gdzie typ to:
INTEGER, LONG, SINGLE, DOUBLE, STRING

Przykłady:

DIM alfa, beta, x, y **AS DOUBLE**

DIM zmienna1 AS LONG

Przykład programu z deklaracjami DIM:

- **DIM** a AS INTEGER
  DIM b AS LONG
  DIM c AS DOUBLE
  CLS
  a = 5
  b = 10
  c = 5#
  PRINT b / c

II sposób deklarowania

W QBasic można deklarować typ zmiennych poprzez podanie tylko pierwszej litery nazwy zmiennej. np. **DEFINT K, S**
Wszystkie zmienne zaczynające sie na zadeklarowana literę (tu K i S) będą tego samego typu (tu INTEGER). Interpretator odrzucił dalsze litery w nazwach i ponadto ustawił pierwsze litery w kolejności alfabetycznej.
DEFINT jest skrótem od define integer i informuje QBasic (interpretator lub kompilator), ze zmienne wypisane po deklaracji **DEFINT** maja być typu całkowitego (integer) - o długości 2 bajtow.
Analogicznie można definiować zmienne innych typów, np.
DEFLNG - zmienne całkowite długie, o długości 4 bajtow.
np. **DEFLNG L-N** - wszystkie zmienne o nazwach zaczynających sie na litery z przedziału L do N (L, M, N lub l, m, n) bedą traktowane jako zmienne typu długich liczb calkowitych.
DEFSNG - deklaracja zmiennych rzeczywistych 4 bajtowych (single precision - pojedynczej dokładności), np. **DEFSNG O-R** lub **DEFSNG O, P, R**
DEFDBL - double precision - podwójnej dokładności np. **DEFDBL S-U**
DEFSTR - deklaracja zmiennych znakowych - typu STRING - (lancuch)
np. **DEFSTR W-Z**

Przykładowy program z **definicja typów**

```
DEFINT K, S
 PRINT start, KONIEC
'wydruk na ekranie zawartości zmiennych
 start = 10
'przypisanie wartości do zmiennej start
 KONIEC = 50          ' -"-                    koniec
 PRINT start, KONIEC
```

III sposób deklarowania

- **Sposób** deklarowania przez zakończenie nazwy zmiennej odpowiednim symbolem:
  % **dla zmiennych INTEGER** - całkowite 2 bajtowe
  & **-"-** **LONG** INTEGER - całkowite 4 bajtowe
  ! **-"-** **SINGLE** PRECISION - rzeczywiste pojedynczej dokładności 4 bajtowe
  # **-"-** **DOUBLE** PRECISION - rzeczywiste podwójnej dokładności 8 bajtowe
  $ **-"-** **STRING** VARIABLE - zmienne typu znakowego
- Uwaga!
  Zmienna, której nazwa nie jest ujęta w żadnej deklaracji i nie jest zakończona żadnym ze znaków (%, &, !, # czy $), jest przez interpretator deklarowana jako zmienna typu ***SINGLE PRECISION*** *- rzeczywista, pojedynczej dokładności, 4 bajtowa*.

Przykład ABCQ7.BAS

- Przykład

```
REM ABCQ7.BAS
‘ Porównywanie liter
CLS
INPUT "Wpisz literę "; lit1$
INPUT "Wpisz literę "; lit2$
IF lit1$ < lit2$ THEN
  PRINT lit1$, lit2$
ELSE
 PRINT lit2$, lit1$
END IF
```

Zmienne znakowe

- **Stale typu znakowego mogą mieć długość od 0 do 32767 bajtów**.
  Aktualna liczba bajtów zajmowana w pamięci operacyjnej przez zmienna typu znakowego zależy od długości przypisanej do niej stałej.
  Jeśli w programie mamy np. 2 instrukcje
  nazwisko1$ = "Roman Goc"
  nazwisko2$ = "Sebastian Zajaczkowski"
  to każda ze zmiennych zajmuje w pamięci operacyjnej liczbę bajtów odpowiednia do długości przypisanej do niej stałej znakowej (tu nazwiska).
  Zmienne rozpoczynające sie na określona literę mogą być w programie deklarowane wielokrotnie, za każdym razem jako innego typu.
  Każda nowa deklaracja anuluje poprzednią.
  Ponadto w momencie zmiany typu danej zmiennej, jej wartość jest zerowana.
  Takie metody deklarowania stosuje sie przy pisaniu bardzo długich programów, kiedy zaczyna brakować miejsca w pamięci komputera.
  Początkującym *programistom nie poleca sie jednak wielokrotnego deklarowania tych samych zmiennych.*

Złożone typy danych

Typy złożone to **tablice** oraz **rekordy**.
Muszą one koniecznie być deklarowane.

Tablice (ang.:arrays):

Przykład zadeklarowania tablicy **jednowymiarowej** czyli ciągu o nazwie "**a**" i 100 elementach:

DIM a(1 TO 100) AS INTEGER

albo: DIM a%(1 TO 100)

Gdzie symbol **%** zastępuje wyraz **integer**.
Każdy ze 100 elementów może przechowywać inną wartość na przykład można podstawić:

a%(1) = 10: a%(2) = 21: a%(3) = 35

Deklaracje tablic dwu-wymiarowych

- A oto przykład **deklaracji tablicy dwu-wymiarowej** w matematyce zwanej **macierzą**

  DIM a$(1 TO 4, 1 TO 3)

Posiada ona 3x4 czyli 12 elementów znakowych (napisy) i każdy może mieć inną wartość:

a$(1,1)	a$(1,2)	a$(1,3)
a$(2,1)	a$(2,2)	a$(2,3)
a$(3,1)	a$(3,2)	a$(3,3)
a$(4,1)	a$(4,2)	a$(4,3)

- Na tablicach operujemy przy pomocy pętli FOR .. NEXT .. lub innych

Wprowadzanie danych do programu

- Najbardziej elementarnym sposobem wprowadzania danych jest wpisywanie ich za pomocą klawiatury.
  QBasic posiada instrukcje **INPUT**, przewidziana do tego celu.
  By zapoznać sie ze sposobem posługiwania sie ta instrukcja wpisujemy do edytora programu QBasic **INPUT** i naciskamy klawisz **F1.**
  Naciskamy dwukrotnie klawisz **F6** i za pomocą klawiszy kierunkowych możemy obejrzeć cały tekst pomocniczy dla słowa **INPUT**.
- **INPUT** reads input from the keyboard or a file.
  LINE INPUT reads a line of up to 255 characters from the keyboard or a file.

INPUT

- Wiersz
  INPUT [;] [„zgłoszenie"{; | ,}] listazmiennych
  należy rozumieć tak:
- **INPUT** jest elementem, który musi wystąpić w instrukcji,
  po słowie INPUT możemy, ale nie musimy wpisać średnik (;), następnie możemy ale nie musimy wpisać dowolny tekst zamknięty cudzysłowami i zakończony średnikiem (;) albo przecinkiem (,), na końcu musi sie znajdować **nazwa zmiennej** lub **kilka nazw** rozdzielonych przecinkami.

Instrukcja **INPUT** ma postać:

INPUT "*tekst żądania danych*"; *zmienna*

- np. INPUT "podaj wartość siły:"; sila
- lub INPUT "siła="; sila
- Tekst zawarty w cudzysłowach będzie wyświetlony aby użytkownik wiedział, że ma wpisać odpowiednią wartość.
  Wpisana wartość zostanie wstawiona do zmiennej, której nazwę podano w tej instrukcji.

Przykłady **INPUT**

Najprostsza instrukcja czytania danych z klawiatury ma postać: **INPUT a12**

Wykonanie tej instrukcji spowoduje wyświetlenie na ekranie znaku zapytania (?) i program będzie czekał na wpisanie z klawiatury stałej, która ma być przypisana do zmiennej o nazwie a12.

Wpisanie stałej kończymy wciśnięciem klawisza Enter.

Jeśli wpisana z klawiatury **stała** nie jest tego typu co zmienna, do której ma być przypisana, to możliwe są 2 przypadki:

a) nastąpi konwersja typu stałej do typu zmiennej i przypisanie tej stałej do zmiennej a12

b) wykonanie programu zostanie zatrzymane z powodu niemożności dokonania konwersji typu.

Przykłady INPUT c.d.

- Instrukcja
  INPUT liczba1, liczba2, liczba3
  zostanie wykonana po wpisaniu z klawiatury 3 liczb rozdzielonych przecinkami i wciśnięciu Enter. W instrukcji INPUT powinno sie zawsze umieszczać informacje dla operatora, jakie wielkości ma wprowadzić.
- Przykładem może być:
  INPUT "Podaj 3 liczby z przedziału 0 do 100"; n1, n2, n3

LINE INPUT

- Instrukcja **LINE INPUT** służy do wprowadzenia **jednej stałej znakowej, która może zawierać w sobie przecinki.**
  W instrukcji INPUT przecinek jest znakiem oddzielającym stale i nie może być wczytany do zmiennej.
  W instrukcji LINE INPUT dopiero [Enter] jest znakiem końca stałej i równocześnie sygnałem do jej przypisania do zmiennej umieszczonej w tej instrukcji.
- Np. instrukcja
  LINE INPUT "Podaj imiona rodzeństwa "; imiona$
  umożliwia przypisanie do zmiennej znakowej imiona$ kilku grup znaków (np. imion) rozdzielonych przecinkami.
  QBasic jest tak napisany, ze wpisanie z klawiatury stałej, która nie może być przypisana do zmiennej, umieszczonej w instrukcji INPUT lub LINE INPUT, nie zatrzymuje wykonywania programu, tylko powrót do początku wykonania instrukcji czytania danych.

Kontrola wprowadzania w INPUT

- Przykład:
  INPUT "Podaj liczbe uczniow w klasie "; liczbauczniow%
  i wpisujemy (zamiast liczby) słowo trzydzieści.
  Na ekranie pojawia sie komunikat
  Redo from start
  co znaczy - wykonaj od poczatku i program czeka na ponowne wprowadzenie danej lub danych.

Input do operacji dyskowych

- Instrukcje
  INPUT #filenumber, variablelist
  LINE INPUT #filenumber%, variable$
  służą do wczytania danych znajdujących się w zbiorze zapisanym na dysku. Wymagają one uprzedniego polaczenia programu z danym zbiorem danych.

Przykład

```
OPEN "DANE.TXT" FOR INPUT AS #1 'Otwarcie pliku do odczytu z numerem 1
PRINT "Tekst pliku:"
DO WHILE NOT EOF(1)        'Pętla działa aż do napotkania znacznika końca pliku 1
    LINE INPUT #1, LINIA$  'Wczytuje linię tekstu z pliku
    PRINT LINIA$           'Wyświetla wczytaną linię tekstu na ekranie
LOOP                                   'Koniec pętli
CLOSE #1                               'Zamknięcie pliku
```

DATA i READ - umieszczanie danych w programie

- Jedna z metod wprowadzenia danych do programu jest umieszczenie ich w tym programie na stale.
  Dane umieszcza sie w wierszach rozpoczynających sie słowem kluczowym **DATA.**
  Dane te przypisywane są do zmiennych instrukcja **READ**.

Przykładem takiego rozwiązania jest fragment programu

```
REM  DATA...READ
CLS
DATA 12, 23, 45, 74, 86, 345, 11
READ a1, a2, a3, a4, a5, a6, a7
PRINT A1, a2, a3, a4, a5, a6, a7
```

którego wykonanie spowoduje przypisanie liczb 12, 23,..11 do zmiennych a1, a2, ...a7.

- Dane z instrukcji DATA mogą być czytane tylko raz.
  Dane te można jednak odtworzyć instrukcja **RESTORE**.
  Instrukcja DATA może znajdować sie w dowolnym miejscu programu, na przykład na końcu.

Wyprowadzanie wyników: PRINT, LPRINT

- Dwie podstawowe metody wyprowadzania wyników działania programu to wyświetlenie ich na **ekranie** monitora **PRINT**
  lub **wydrukowanie** na drukarce: **LPRINT**
- Trzeci sposób to zapisanie wyników do **zbioru** na dysku, celem późniejszego zapoznania sie z nimi - wyświetlenia na ekranie i ewentualnego wydrukowania.
  Zapisywanie wyników na dysku jest w wielu przypadkach etapem pośrednim realizacji dużego zadania za pomocą programów komputerowych.
  Jeden program wykonuje jakiś etap obliczeń, zapisuje wyniki do zbioru na dysku, a inny program wczytuje zawartość tego zbioru, która stanowi dla niego dane wejściowe.

Instrukcje wyprowadzania wyników

- **PRINT [lista wyrażeń]** – wyświetlanie
- **LPRINT [lista wyrażeń]** – wydruk na drukarkę
- np.
  PRINT wyn1, wyn2, wyn3 wydruk na ekran
  w której wyn1, wyn2, wyn3 to nazwy zmiennych, w których zapamiętane zostały wyniki działania programu.
  Wydrukowanie wymaga instrukcji
  LPRINT wyn1, wyn2, wyn3 – na drukarkę
  wymaga drukarki podłączonej do komputera, lub włączonej do sieci i odpowiednio przygotowanej
- Zamiast **PRINT** można pisać znak **pytajnika ?**

PRINT USING

- **PRINT USING <wyrażenie** wyświetla łańcuchy lub liczby z użyciem określonego formatu
- Jest to taki wariant instrukcji **PRINT** , który pozwala wyświetlać wyniki w równych słupkach o określonej ilości cyfr przed i po kropce dziesiętnej. Można przetłumaczyć jej znaczenie jako: **"drukuj używając podanego szablonu formatu"**.
- W najprostszym przypadku ten szablon formatu to zawarty w cudzysłowach ciąg tylu znaków # ile cyfr chcemy otrzymać odpowiednio przed i po kropce dziesiętnej.

Formaty łańcuchowe:

- **"!"** - tylko pierwszy znak łańcucha będzie wyświetlany
- print using "\ n spacji\" wyświetlonych będzie n+2 znaków łańcucha. Jeśli n+2 będzie większe od długości łańcucha to z prawej strony dodawane będą spacje
- **"&"** wyświetlony będzie cały łańcuch
- Formaty liczb:

  # reprezentuje każda wyświetlaną cyfrę w tekście

Przykłady PRINT USING

- Przykład 1

```
a = 123.456
PRINT USING "###.##"; a
a$ = "ABCDEFG"
PRINT USING "!"; a$
PRINT USING "\ \"; a$
```

- Przykład 2

```
CLS
PRINT "Obliczanie kwadratów liczb"
PRINT "liczba", "kwadrat"
FOR y = 0 TO 100 STEP 9.05
PRINT USING "######.####"; y; y^2
NEXT y
```

Program QBPR1.BAS

- REM QBPR1.BAS ' Nazwa programu
  REM Wczytywanie liczb, znaków, sumy, wydruk – komentarz odnośnie funkcji programu
  CLS ' Kasowanie ekranu, REM lub apostrof oznacza komentarz
  DEFDBL A-L ' Deklaracja zmiennych A-L jako Long
  DEFSTR P-Z ' Deklaracja zmiennych A-L jako String
  INPUT "Wpisz dowolna liczbe "; lb1 ' Wprowadzenie liczby
  INPUT "wpisz inna liczbe "; lb2
  INPUT "Wpisz znaki "; znak1 ' Wprowadzenie znaku
  INPUT "wpisz znaki "; znak2
  lb3 = lb1 + lb2
  znak3 = znak1 + znak2
  PRINT lb1, lb2, lb3 ' Wydruk na ekranie
  PRINT znak1, znak2, znak3
  PRINT lb1; lb2; lb3
  PRINT znak1; znak2; znak3
  PRINT "lb1="; lb1, "lb2="; lb2, "lb3=lb1+lb2="; lb3
  END ' Koniec programu

Program QBRPR2.BAS

```
REM QBRPR2.BAS
REM Wydruk liczb z uzyciem formatowania
CLS
DEFDBL A-L
a = 100
b = 3
c = a / b
PRINT c, c, c
PRINT
PRINT c; c; c
PRINT
PRINT USING "###.###"; c; c; c
PRINT
PRINT USING "+###.####"; c; c; c
PRINT USING "+########.####"; c; c; c
PRINT USING "+#######.####"; c; c; c
PRINT USING "+##########.####"; c; c;
```

Zapis do pliku

Przykład programu zapisującego wyniki do pliku WYNIKI.TXT:

```
CLS
PRINT "Drukowanie do pliku WYNIKI.TXT"
OPEN „WYNIKI.TXT" FOR OUTPUT AS #1
    PRINT #1, "Obliczanie kwadratow liczb"
    PRINT #1, "liczba", "kwadrat"
    FOR y = 0 TO 100 STEP 9.05
        PRINT #1, USING "######.####"; y; y ^ 2
    NEXT y
CLOSE #1
```

- Sprawdź jego działanie - otwórz plik wyników w NOTATNIKU i przejrzyj.