

LEKCJA11 – KONSOLA - CIĄGI LICZBOWE

W trybie konsoli ekran jest podzielony na pola (80x25), w których można wyświetlić znaki z alfabetu ASCII w 16 kolorach. Nie dysponujemy żadnymi gotowymi obiektami. Możemy jedynie wczytywać liczby do zmiennych, obliczać i wyświetlać wyniki na ekranie. Tryb konsoli to sposób tradycyjnego projektowania programów – sprzed ery języków obiektowych i graficznych systemów okienkowych.

Instrukcje wejścia i wyjścia

By działanie programu miało jakiś sens, musi oczywiście przyjmować dane, przekształcać je i wyprowadzać wyniki. Przyjmowanie danych realizujemy za pomocą **instrukcji wejścia - READLN** wprowadzając dane za pomocą klawiatury. Wyprowadzanie wyników na ekran, realizujemy za pomocą **instrukcji wyjścia - WRITELN**.

Przykłady zastosowania obu instrukcji przedstawiono poniżej:

writeln('Pascal'); na ekranie napis Pascal i kursor przeskoczy do nowego wiersza
write('Pascal'); na ekranie napis Pascal i kursor nie przeskoczy do nowego wiersza
writeln; nowy wiersz - kursor przeskakuje do nowego wiersza

write('Ala '); Po wykonaniu tych trzech instrukcji -
write('ma '); na ekranie napis "Ala ma kota" w jednym wierszu
writeln('kota'); i kursor przeskoczył do nowego wiersza

writeln('Ala ','ma ','kota'); jak poprzednio - instrukcja ma 3 parametry

x:='Ala ';
writeln(x,'ma kota'); jak poprzednio - wypisana zawartość zmiennej x

readln; program oczekuje na wciśnięciu klawisza ENTER
readln(liczba); program oczekuje na wpisanie informacji z klawiatury (mruga kursor). Po wpisaniu danych naciskamy klawisz ENTER i w tym momencie dane zostają przypisane do zmiennej. Mogą to być teksty lub liczby, w zależności od typu zmiennej.

Formatowania wyprowadzanych na ekran napisów

Podczas wyprowadzania na ekran liczb rzeczywistych, np. **writeln(10/3);** komputer formatuje liczbę w postaci wykładniczej 3.333333333E+00. Istnieje sposób, aby ta liczba była napisana bardziej „zrozumiale”. Jeśli np. instrukcja będzie wyglądała tak: **writeln(10/3:8:3);** na ekranie pojawi się: 3.333. Dodatkowe liczby 8 i 3 oznaczają, że na wynik działania rezerwujemy 8 miejsc, w tym 3 na część ułamkową. Jeśli chcemy bez miejsc po przecinku - wpisujemy np. **writeln(10/3:8:0);** Jeśli z dwoma miejscami po przecinku, przyklejone do lewej krawędzi: **writeln(10/3:0:2);**

APLIKACJA KONSOLOWA

Aby uruchomić Delphi w trybie programowania konsoli należy z menu wybrać:

FILE – NEW – OTHER i w okienku wybrać CONSOLE APPLICATION.

Pojawia się najprostsza z możliwych struktura początkowa programu. Przed

słowem BEGIN deklarujemy zmienne i definiujemy procedury. Pomędzy słowami BEGIN i END realizujemy nasz program.

PROGRAM konsola;
USES SysUtils;
BEGIN
END.

PRZYKŁAD – LEKCJA111 – ciąg fibbonaciego

- wczytujemy z klawiatury liczbę do zmiennej X
- w pętli FOR
 - * wyświetlamy na ekranie w jednym wierszu zmienną I i sumę f1 i f2
I na 3 polach i sumę na 6 polach
 - * zapamiętujemy zmienną F2 w F
 - * w zmiennej F2 zapamiętujemy sumę zmiennych F1 i F2
 - * w zmiennej F1 zapamiętujemy F

Program wyświetla na ekranie X kolejnych liczb ciągu Fibbonaciego.

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,...

f1:=0;
f2:=1;
write('podaj liczbę: ');
readln(x);
for i:=1 to x do
begin
writeln(i:3,f1+f2:6);
f:=f2;
f2:=f1+f2;
f1:=f;
end;

CIĄGI LICZBOWE - ĆWICZENIA

Za pomocą instrukcji WRITELN i poznanych na poprzednich lekcjach instrukcji:

- przypisania
- arytmetyczne (+ - * / ^ DIV MOD), relacji (= <> < > <= >=, logiczne (NOT OR AND XOR)
- pętle iteracyjna i logiczne: FOR, REPEAT, WHILE
- warunkowa i wyboru: IF THEN ELSE, CASE
- złożona BEGIN END

DIV i MOD

x DIV y – ile razy liczba y mieści się w liczbie x, np. $14 \text{ div } 4 = 3$

x MOD y – jaka jest reszta z dzielenia całkowitego liczby x przez y, np. $14 \text{ mod } 4 = 2$

OPERATORY LOGICZNE

we wszystkich przykładach na początku zmienna x:=5;

NOT – zaprzeczenie, np. IF NOT(x=5) THEN writeln(0) ELSE writeln(1); *(na ekranie wynik 1)*

OR – suma logiczna, np. IF (x<=5) OR (x>7) THEN writeln(0) ELSE writeln(1); *(na ekranie wynik 0)*

AND – iloczyn logiczny, np. IF (x<=5) AND (x>7) THEN writeln(0) ELSE writeln(1); *(na ekranie wynik 1)*

XOR – różnica symetryczna, np. IF (x<=5) XOR (x>7) THEN writeln(0) ELSE writeln(1); *(na ekranie wynik 0)*

XOR prawdziwa, gdy tylko jedno ze zdań jest prawdziwe

A) Napisz programy, które będą na ekranie wypisywały podane poniżej ciągi liczbowe.

Do każdego pytania powinny być utworzone trzy programy każdy z inny pętlą: FOR, WHILE, REPEAT realizujące to samo zadanie.

- 1) 10 kolejnych liczb naturalnych
- 2) 10 kolejnych liczb naturalnych licząc wstecz od 10
- 3) 10 kolejnych liczb parzystych
- 4) 10 kolejnych liczb nieparzystych
- 5) 10 kolejnych liczb dla następującego ciągu 0,6,12,18,24...
- 6) 10 kolejnych liczb dla następującego ciągu 24,18,12,6,0...
- 7) 10 kolejnych liczb dla następującego ciągu 1,1,2,4,7,11...
- 8) 10 kolejnych liczb dla następującego ciągu 1,2,3,5,8,13,21...
- 9) 10 kolejnych liczb dla następującego ciągu 0,1,3,4,6,7,9...
- 10) 10 kolejnych liczb dla następującego ciągu 1,4,9,16,25...

B) Jakie liczby wypiszą na ekranie poniższe programy:

- 1) for i:=1 to 5 do writeln(i mod 2);
- 2) for i:=1 to 5 do writeln(i div 2);
- 3) i:=1; while i<5 do begin writeln(i+i); i:=i+1 end;
- 4) i:=3; while i<=5 do begin writeln(i-i); i:=i+1 end;
- 5) i:=5; repeat writeln(i); i:=i-1 until i=1;
- 6) i:=1; repeat i:=i+1; writeln(i+10) until i=5;
- 7) for i:=1 to 10 do if (i>2) and (i<=5) then writeln(i);
- 8) i:=3; while (i>2) and (i<=5) do begin writeln(i); i:=i+1 end;
- 9) i:=3; repeat writeln(i); i:=i+1 until (i<3) or (i>7);

PRZYKŁADOWE ROZWIĄZANIE

A 1

FOR i:-1 TO 10 DO WRITE(i, ' '); jeśli jedna liczba obok drugiej

FOR i:-1 TO 10 DO WRITELN(i); jeśli jedna liczba pod drugą

i:=1; REPEAT WRITELN(i); i:=i+1; UNTIL i=11;

i:=1; WHILE i<=10 DO BEGIN WRITELN(i); i:=i+1 END;

B 1

1 0 1 0 1