## LEKCJA 4 - PROCEDURY

Z punktu widzenia użytkownika, praca z programami odbywa się według schematu:

- dostarczyć dane do programu (klawiatura, myszka, skaner itp.)
- przetworzyć informacje w programie
- zapamiętać wyniki działania programu

### Programisty zadaniem jest:

- <u>opracować komunikację</u> pomiędzy użytkownikiem, a programem, tzn. jak użytkownik ma dostarczać dane i w jakiej formie będzie otrzymywał wyniki,
- <u>sformułować metodę</u>, zgodnie z którą program będzie działał (schematycznie podany sposób rozwiązania problemu nosi nazwę **algorytm**),
- opisać algorytm w konkretnym języku programowania,
- <u>sprawdzić poprawność</u> działania programu.

### Procedury

Jeśli mamy do wykonania jakieś skomplikowane zadanie, lepiej jest podzielić je na mniejsze fragmenty. W programowaniu również pewne elementy naszej pracy możemy wydzielić, po to, żeby łatwiej było je zaprojektować i żeby program miał czytelniejszą strukturę. W ObjectPascal'u praktycznie cały program, to zestaw takich wydzielonych elementów - **PROCEDUR**. Procedury związane są z reguły z jakimś obiektem i zdarzeniem. Jeśli przejrzymy kod źródłowy dowolnego programu, to zauważymy, że informacja o procedurze pojawia się w dwóch miejscach: **nagłówki procedur** zapisywane są na początku, w części oznaczonej TYPE. Same **procedury zawierające konkretne instrukcje** zdefiniowane są po słowie IMPLEMENTATION. Startowy kod procedury (nagłówek, początek i koniec) tworzony jest automatycznie przez np. podwójne kliknięcie w obiekt lub w zdarzenie z nim związane. Procedury można też tworzyć ręcznie, jak w przykładzie LEKCJA041. Parametry procedur są jakby "drzwiami", przez które przekazujemy do procedury informacje.

### PRZYKŁAD - LEKCJA041

Ręcznie definiujemy procedurę, która narysuje dowolny prostokąt Położenie prostokąta zależne jest od punktu (X,Y). Wymiary: szerokość – S i wysokość – W, kolor - K.

- przejść do okna kodu źródłowego (UNIT)
- w części kody oznaczonej TYPE, przed słowem PRIVATE wpisać nagłówek
- po słowie IMPLEMENTATION wpisać procedurę właściwą

# Aby program narysował jakikolwiek prostokąt należy **wywołać procedurę** i **przypisać ją do jakiegoś zdarzenia.**

Wywołujemy za pomocą jej nazwy uzupełnionej liczbowymi parametrami, np.: **PROST(300,100,200,50,clRed);** 

Przypisujemy np. do zdarzenia ONCLICK w umieszczonym na formatce przycisku (BUTTON) lub do zdarzenia ONRESIZE formatki – rysowanie automatycznie po uruchomieniu programu.

### Za pomocą przycisku

- na formatce narysować przycisk BUTTON
- podwójnie kliknąć w przycisk procedura dla zdarzenia ONCLICK o nazwie BUTTON1CLICK
- wpisać **PROST**(**300,100,200,50,clRed**);
- uruchomić aplikację i kliknąć w przycisk

### Zaraz po uruchomieniu

- podwójnie kliknąć w pole ONRESIZE w okienku właściwości formatki
- wpisać **PROST(100,100,200,50,clRed)**;

#### interface uses Windows, Messages, SysUtils, Variants, Class Dialogs, StdCtrls; TForm1 = class(TForm) Button1: TButton; procedure prost(x,y,w,s,k:integer); procedure Button1Click(Sender: TObject); procedure FormResize(Sender: TObject); private ( Public declarations ) end; var Form1: TForm1; implementation (\$R \*.dfm) procedure TForm1.prost(x,y,w,s,k:integer); begin with canvas do begin begin pen.color:=k; moveto(x,y); lineto(x+s,y); lineto(x+s,y+w); lineto(x,y+w); lineto(x,y+w); end; d:



procedure prost(x,y,w,s,k:integer);

procedure TForm1.PROST
 (x,y,w,s,k:integer);
begin
 with canvas do
 begin
 pen.color:=k;
 moveto(x,y);
 lineto(x+s,y);
 lineto(x+s,y+w);
 lineto(x,y+w);
 lineto(x,y);
 end;
end;

