

```
%% Przykładowe zadanie wykładowe zwracające macierz 'wymiar' na 'wymiar' elementów,%
%% będących wyznaczoną funkcją z argumentu: sumy kolumny i wiersza bieżącego tej macierzy%
%% Przykładowe wywołanie funkcji zadaniowyk:
%%-----
% [A] = zadaniowyk_dowolna_funkcja_adds_mults(3,0.001,'sin+cos*cos'); %<= policz macierz 3x3 elem.,będących sinusem z arg. sumy
wiersz i kolumny w tej mac%
function [A] = zadaniowyk_dowolna_funkcja_adds_mults(wymiar,dkl,fn)
A = zeros(wymiar,wymiar);
for i = 1 : wymiar,
    for j = 1 : wymiar,
        A(i,j)= dowolnafunkcjadokl_adds_mults(0,i+j,dkl,fn);
    end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Przykład wywołania całkowania ze skryptem nadrzędnym z użyciem dowolnej prostej(jednokrotnej) funkcji:%
%-----%
% pole=dowolnafunkcjadokl_adds_mults(0,1,0.001,'sin+sin') %<=wyznacz pole - całk. oznaczoną pod funkcją sinus w granicach [0,1
radian]%
% pole=dowolnafunkcjadokl_adds_mults(0,1,0.001,'cos*tan+sin') %<=wyznacz pole - całk. oznaczoną pod funkcją sinus w granicach [0,1
radian]%
% pole=dowolnafunkcjadokl_adds_mults(0,1,0.001,'exp*tan+ln*cos') %<=wyznacz pole - całk. oznaczoną pod funkcją eksponent(x) w
granicach [0,1radian]%
function [pole] = dowolnafunkcjadokl_adds_mults(a,b,dkl,fn) %przykładowo z wywołaniem: dowolnafunkcjadokl(0,1,0.001,'sin');%
pole = 0;      % pole inicjuj na zerowe
n = 5;         % zacznij od 5 podprzedziałów
rozn_wynik = 10; % zmienna określ. nieco dalej różnicę bezwzgl. wyników: ostatniego z przedostatnim%
szr = abs(b-a)/n; % wyznacz szerokość każdego z podprzedziałów
%figure,      % przygotuj (tzn. otwórz) okno wydruku graficznego danych
while(dkl < rozn_wynik) % dopóki szacowany błąd nie jest mniejszy niż deklarowany margines błędu%
%   OsX = a:szr:b;      % wyznacz WEKTOR argumentu całego odcinka osi X w przedz.[a,b] %
%   Wielomian = mfv1_adds(fn,OsX); % wyznacz WEKTOROWO przeciw-argument osi X (wartość funkcji)
%   plot(OsX,Wielomian,'r');% wstępnie na czerwono wyświetl linią ciągłą funkcję
%   hold on;           % przytrzymaj uchwyt okna wyświetleń graficznych danych
%   bar(OsX,Wielomian,0.975,'b'); % dorysuj na wyk. 'dość zgrubnie' szereg słupków-prost., przymierzanych do krzywej%
%   disp('Proszę wprowadzić: ''return'' albo sprawdzić np. wartości zmiennych roboczych :');
%   keyboard;          % przekaż sterowanie do do promptu użytkownika: K>>_
%   hold off;          % zwolnij uchwyt okna graficznego, by w nast. pętli odświeżyć wykres
pole_przedostatnie = dowolnafunkcja_adds_mults(a,b,n,fn); % policz pole - całki oznaczonej zgrubnie
n=2*n              % zdubluj liczbę podprzedziałów
szr=szr/2;         % zmniejsz odpowiednio ich szerokości
pole_ostatnie      = dowolnafunkcja_adds_mults(a,b,n,fn); % a obecnie: policz pole - całki oznaczonej dokładniej!
rozn_wynik         = abs(pole_ostatnie - pole_przedostatnie); % szacuj margines błędu
end;
pole = pole_ostatnie; % de facto: pole ostatnie, to te najdokładniejsze!%Przykład wywołania całkowania ze skryptem nadrzędnym z
użyciem dowolnej prostej(jednokrotnej) funkcji:%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
%-----%
%Przykład wywołania całkowania z użyciem dowolnej prostej(jednokrotnej) funkcji:%
%-----%
% pole=dowolnafunkcja_adds_mults(0,1,10000,'sin*sin+cos*cos') %<=wyznacz pole - całk. oznaczoną pod funkcją sinus w granicach [0,1
radian]
function [pole] = dowolnafunkcja_adds_mults(a,b,n,fn) % wersja przybliżania trapezami pola, w reprezent. całki ozn.
pole = 0;      % inicjuj wartość pola całki oznaczonej y=x^2+x
szr = abs(b-a)/n; % wyznacz szer. każdego z podprzedz., dzieląc długość przedziału [a,b] na n pododcinków
XL=a;          % przypisz wart. lewej granicy podprzedziału - XL - lewą granicę przedziału:[a,b]
XP=a;          % przypisz wart. prawej granicy podprzedziału - XP - prawą granicę przedziału [a,b]
XSR=a;         % srodek każdego z podprzedziału - na razie domyślnie ustawiony na wartość a
for i=1:n      % w pętli n-krotnej dosumowywuj pola prostokątów podprzedziałów o podstawie: 'szr'
    XP = XP + szr; % przed właściwym doliczeniem pola trapezu - ustaw wart. prawej granicy podprzedziału
    XSR = (XL+XP)/2;% określ położenia środka podstawy trapezu (na osi argumentu X)
    pole = pole + szr*abs(mfv1_adds(fn,XSR)); %dosumuj w pętli pole jednego trapezu w podprzedz.[XL,XP]
    XL = XP + szr; % przed końcem każdego przebiegu pętli 'for' przesun XP o szer. 1 podprzedz.
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [mval_adds] = mfv1_adds(mstrng,args)
mpos = strfind(mstrng,'+');
if isempty(mpos) == 0
% disp(['W łańcuchu: ''' mstrng ''' operator '+' został znaleziony na pozycji: ' num2str(mpos(1))]);
else
% disp(['W łańcuchu: ''' mstrng ''' brak jest operatora '+' ']);
end;
switch isempty(mpos)
    case 0 % dokonuj dalszego podziału łańcucha do evaluacji rekurencyjnie z udziałem funkcji mfv1_adds
        mstrng(1:mpos(1)-1);
        mstrng(mpos(1)+1:end);
        mval_adds = mfv1_adds(mstrng(1:mpos(1)-1),args) + mfv1_adds(mstrng(mpos(1)+1:end),args);
    case 1 % szacuj wyrażenie pozbawione już operatora '+' z pomocą innej podrzędnej funkcji rekurencyjnej
        mval_adds = mfv1_mults(mstrng,args);
        return;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [mval_mults] = mfv1_mults(mstrng,args)
mpos = strfind(mstrng,'*');
if isempty(mpos) == 0
% disp(['W łańcuchu: ''' mstrng ''' operator '*' został znaleziony na pozycji: ' num2str(mpos(1))]);
else
% disp(['W łańcuchu: ''' mstrng ''' brak jest operatora '*' ']);
end;
switch isempty(mpos)
```

```
case 0    % dokonuj dalszego podziału łańcucha wyrażenia
    mstrng(1:mpos(1)-1);
    mstrng(mpos(1)+1:end);
    mval_mults = mfvl_mults(mstrng(1:mpos(1)-1),args) .* mfvl_mults(mstrng(mpos(1)+1:end),args);
case 1    % szacuj wyrażenie pozbawione już operatora '*'
    mval_mults = feval(mstrng,args);
return;

end;
```