

```

:- use_module(library(jpl)).

demo_draw :-
    jpl_new('ShapeDrawingFrame', ['*** MyTitleHere ***'], Frame),
    jpl_call(Frame, setSize, [1000, 1200], _),

    add_black_background(Frame),

    draw_object(Frame, 400, 300),

    % display Frame
    jpl_call(Frame, setVisible, [@(true)], _),

    save_drawing(Frame).

% +Frame: der ShapeDrawingFrame dem ein schwarzer Hintergrund hinzugefügt
% werden soll
% beispielhaftes Hinzufügen einer Shape zu einem ShapeDrawingFrame.
add_black_background(Frame) :-
    % Größe des Frames herausfinden
    jpl_call(Frame, getSize, [], Dimension),
    % Ausdehnung von Dimension auslesen
    jpl_get(Dimension, height, Height),
    % ... alternativ getter benutzen
    jpl_call(Dimension, getWidth, [], Width),
    % beachten Sie, dass innere Klassen nicht mit einem Punkt als Trenner,
    % sondern mit einem Dollarzeichen getrennt werden: im Paket
java.awt.geom
    % gibt es eine Klasse Rectangle2D mit einer inneren Klasse Float.
    % Ähnliches gilt für viele Klassen im Geometry-Paket.
    jpl_new('java.awt.geom.Rectangle2D$Float', [0, 0, Width, Height],
Rectangle),
    % java.awt.Color enthält einige vordefinierte Farben:
    jpl_get('java.awt.Color', black, Black),
    jpl_call(Frame, addFillShape, [Rectangle, Black], _).

% draw_object(Frame,Size,CurrentSize,*** add additional parameters here, if
% needed ***)
% recursively adds some shapes of size Size to Frame
% CurrentSize is decreased recursively fom Size to 0
draw_object(_,_,0 /* , *** add additional parameters here, if needed *** */).
draw_object(Name,Size,CSize /* , *** add additional parameters here, if
% needed *** */) :-
    CSize > 0 ,           % only for positive integers

    % *** insert the computation of graphical parameters here ***

    % *** create and add the current graphical object here ***

```

```

    % decrement CurrentSize and call draw_object recursively
    CSizeNew is CSize - 2,
% writeln(CSizeNew),
    draw_object(Name,Size,CSizeNew).

% save drawing in the frame (if requested by user)
save_drawing(Frame):-
    % if desired save the display as .png
    write_ln('Save the graphics (y/n): '),
    get_single_char(A),
    put_code(A),nl,
    (A==121 ->
        (write_ln('enter file name (without extension): '),
            read_line_to_codes(user_input,X),
            atom_codes(File,X),
            atom_concat(File,'.png',FileName),
            jpl_call(Frame, storeScreenShot, [FileName], _) ) ; true).

% this directive runs the above .
:- demo_draw.

```