

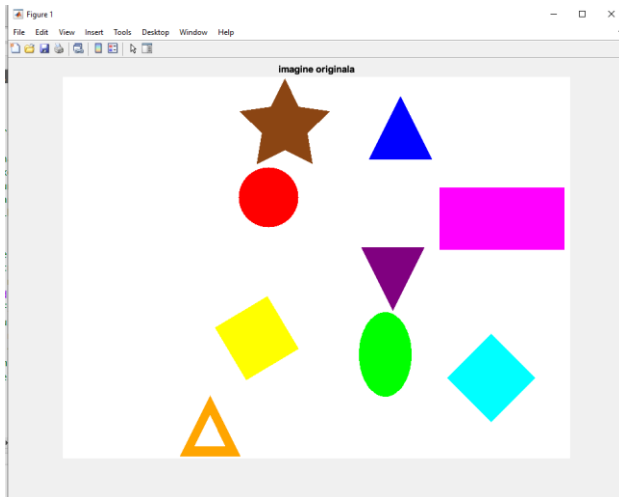
Tema 2 - Reunoasterea formelor

1. Am citit, am afisat si am stocat in variabila img imaginea 1305B_1306A.png

```
img = imread('1305B_1306A.png');
```

```
imshow(img);
```

```
title('imagine originala');
```



2. Am facut conversia la tonuri de gri si binarizarea

```
gray = rgb2gray(img);
```

```
bw = imbinarize(gray, 0.9);
```

%binarizez imaginea grayscale; o transform in imagine alb-negru

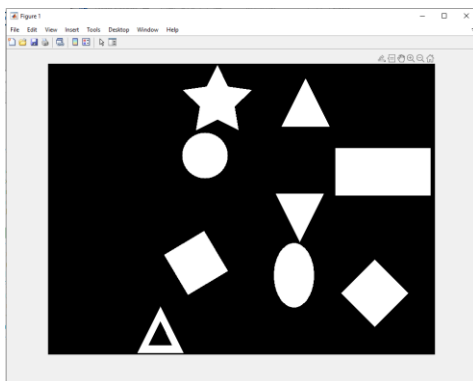
%pixelii din gray cu o intensitate mai mare decat 0.9

% (unde intensitatile sunt normalizate la intervalul [0, 1]) vor fi setati la 1 (alb) în bw,

% iar cei cu o intensitate mai mica sau egala vor fi setati la 0 (negru).

```
bw = ~bw; % inversez bw: formele sa fie albe, iar fundalul negru
```

```
imshow(bw);
```



3. Etichetez si extrag caracteristicile pentru fiecare forma identificata

```
[L, num] = bwlabel(bw);
```

```
stats = regionprops(L, 'Centroid', 'Area', 'Perimeter', 'BoundingBox',  
'Eccentricity');
```

%Calculeaza proprietatile pentru fiecare regiune etichetata din matricea L:

%centroid reprezinta coordonatele centrului fiecarei forme detectate;

%aria (area) este numarul de pixeli care compun fiecare regiune;

%perimetrul (perimeter) indica lungimea conturului fiecarei regiuni;

%bounding box este cel mai mic dreptunghi care incadreaza complet fiecare regiune;

%excentricitatea (eccentricity) elipsei masoara cat de alungita este forma regiunii respective.

4. Parcurgerea fiecarei forme si analiza ei

```
for k = 1:num
```

```
    mask = (L == k);
```

```
    B = bwboundaries(mask);
```

```
    boundary = B{1};
```

%incepe parcurgerea fiecarei forme detectate creand o masca binara conform conturului gasit

5. Simplificare contur si identificare numar de varfuri

```
    simplificare = reducepoly(boundary, 0.03);
```

% reduce nr de puncte de pe contur pentru a aproxima varfurile; 0.3 este gradul de simplificare

```
    num_varfuri = size(simplificare, 1) - 1;
```

% numarul de varfuri este nr de randuri din conturul deja simplificat minus 1 (am sczut 1 pentru ca punctul final sa coincida cu cel final si sa obtin o forma inchisa)

6. Calculare caracteristici geometrice

```
    area = stats(k).Area;
```

% obtine aria formei din structura 'stats'

```
    perimeter = stats(k).Perimeter;  
%stats'
```

% obtine perimetrul formei curente din structura

```
    circularity = (4 * pi * area) / (perimeter^2); % calculeaza circularitatea
```

```
    bb = stats(k).BoundingBox;
```

% obtine dreptunghiul care incadreaza forma

```
    centroid = stats(k).Centroid;
```

% obtine centrul formei

Barbu Oana Gabriela
grupa 1306A

ratio = bb(3) / bb(4); % calculeaza raportul latime/inaltime al
dreptunghiului care incadreaza forma

7. Clasificarea formei

if num_varfuri == 3 % daca forma are 3 varfuri este probabil sa fie triunghi
shape = 'triunghi';
elseif num_varfuri == 4 % daca forma are 4 varfuri ar putea fi patrat, dreptunghi
sau romb

% analizez laturile si unghiurile cu ajutorul conturului simplificat

pts = simplificare(1:4, :); % extrag cele 4 varfuri
v1 = pts(2,:) - pts(1,:); % vector pentru prima latura
v2 = pts(3,:) - pts(2,:); % vector pentru a 2a latura
v3 = pts(4,:) - pts(3,:); % vector pentru a 3a latura
v4 = pts(1,:) - pts(4,:); % vector pentru a 4a latura

%calculez lungimile laturilor

l1 = norm(v1);
l2 = norm(v2);
l3 = norm(v3);
l4 = norm(v4);
ang1 = acosd(dot(v1, v2) / (l1 * l2)); % calculeaza unghiul dintre prima și a
2a latura
ang2 = acosd(dot(v2, v3) / (l2 * l3)); % calculeaza unghiul dintre a 2a si a
3a latura

if abs(ang1 - 90) < 10 && abs(ang2 - 90) < 10 % daca unghiurile sunt apropiate de
90 de grade

if abs(ratio - 1) < 0.15 % daca raportul latime/inaltime este apropiat de 1 ar
trebuie sa fie patrat

shape = 'patrat';
else
shape = 'dreptunghi';

Barbu Oana Gabriela
grupa 1306A

```
end

else % daca unghiurile nu sunt apropiate de 90 de grade este probabil sa
fie un romb
    shape = 'romb';
end

elseif num_varfuri >= 10 && circularity < 0.65
% daca are multe varfuri și circularitatea este mica este probabil o stea
    shape = 'stea';

elseif circularity > 0.7 % daca circularitatea este mare ar putea fi ori cerc ori
elipsa
    if stats(k).Eccentricity < 0.6 % daca excentricitatea este mica ar trebui sa fie un cerc
        shape = 'cerc';
    else
        shape = 'elipsa';
    end
else
    shape = 'alta forma';
end
```

8.Reprezentare rezultat

```
rectangle('Position', bb, 'EdgeColor', 'red', 'LineWidth', 2);
% deseneaza un dreptunghi rosu în jurul formeii

text(centroid(1), centroid(2), shape, 'Color', 'black', 'FontSize', 14,
'FontWeight', 'bold', 'HorizontalAlignment', 'center');
%adauga textul cu numele fiecărei forme

end
```

