# Лабораторная работа № 6. Преобразование объекта на плоскости

## Задача

Построить произвольный объект в первой четверти системы координат (например, флажок; количество вершин объекта должно быть не меньше 7) и организовать следующие преобразования объекта:

* общее преобразование (изменение масштаба, симметричное отражение относительно осей, сдвиг),
* вращение объекта относительно начала координат,
* комбинированное преобразование (преобразование общего вида + вращение на произвольный угол относительно начала координат).

Матрица общего преобразования и угол поворота должны вводиться через пользовательский интерфейс, координаты вершин объекта можно указать явно в коде программы.

## Математическая модель

## Код программы

WD = 500  
ZR = WD / 2  
KF = 10  
FLAG = [(5, 5), (5, 16), (10, 16), (7, 13), (10, 10), (6, 10), (6, 5)]  
  
  
def dec\_to\_scr(cords: tuple):  
 return cords[0] \* KF + ZR, ZR - cords[1] \* KF  
  
  
def axis():  
 cnvs.delete("all")  
 cnvs.create\_line(0, 250, 500, 250, arrow=LAST)  
 cnvs.create\_line(250, 500, 250, 0, arrow=LAST)  
 cnvs.create\_text(WD - 20, ZR + 20, text='X')  
 cnvs.create\_text(ZR - 20, 20, text='Y', )  
 vals = [i for i in range(-50, 51, 5)]  
 cords = [int(ZR + i\*KF) for i in vals]  
 for i in range(len(cords)):  
 cnvs.create\_oval(cords[i], ZR, cords[i] + 2, ZR + 2, fill='black')  
 cnvs.create\_text(cords[i] + 5, ZR + 10, text=str(vals[i]))  
 cnvs.create\_oval(ZR, cords[i], ZR + 2, cords[i] + 2, fill='black')  
 cnvs.create\_text(ZR + 5, cords[i] + 10, text=str(-vals[i]))  
  
  
def transl\_to\_rad(x):  
 return x \* math.pi / 180  
  
  
def transfer():  
 axis()  
 if int(angle.get()) == 0:  
 matrix = ((float(a11.get()), float(a12.get())),  
 (float(a21.get()), float(a22.get())))  
 else:  
 matrix = ((math.cos(transl\_to\_rad(int(angle.get()))), math.sin(transl\_to\_rad(int(angle.get())))),  
 (-math.sin(transl\_to\_rad(int(angle.get()))), math.cos(transl\_to\_rad(int(angle.get())))))  
 new\_flag = list()  
 for each in FLAG:  
 new\_cords = (each[0]\*matrix[0][0] + each[1]\*matrix[1][0],  
 each[0]\*matrix[0][1] + each[1]\*matrix[1][1])  
 new\_flag.append(new\_cords)  
 new\_flag\_scr = [dec\_to\_scr(i) for i in new\_flag]  
 cnvs.create\_polygon(new\_flag\_scr)

## Результаты



