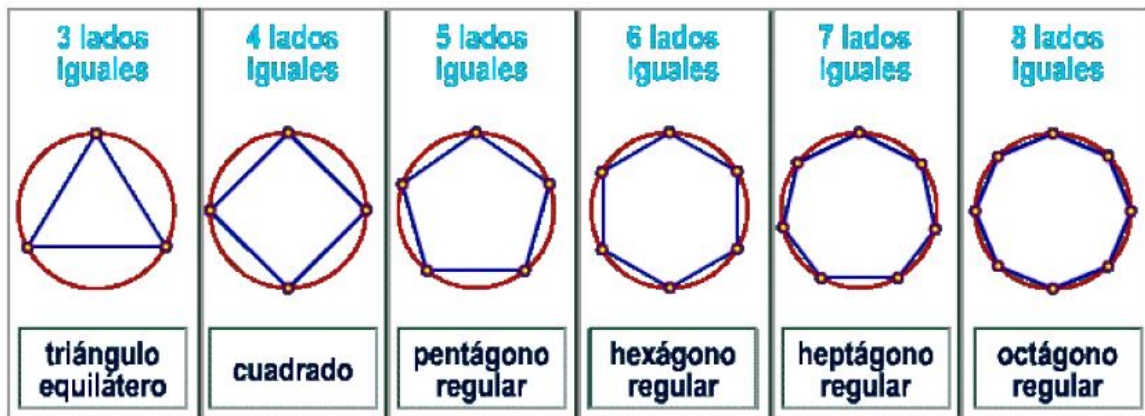




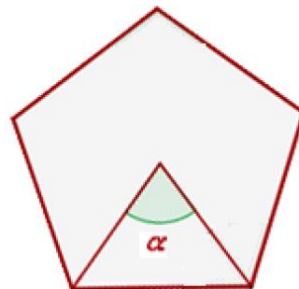
## PROYECTO 2

Elaborado por: M. en C. Ukranio Coronilla

Reutilice el código visto en la clase 6 para derivar una clase que representa a los polígonos regulares, inscritos en una circunferencia de radio variable llamada **PoligonoReg** a partir de la clase **PoligonoIrrreg**.



La clase **PoligonoReg** sólo tendrá como datos adicionales, el número de vértices del polígono regular, y el ángulo  $\alpha$  que se subtiende en cada triángulo que lo compone como muestra la figura:



El constructor de **PoligonoReg** debe recibir como único parámetro el número de vértices, y el único método de la clase será **obtieneArea**, el cual devuelve el área del polígono regular.

Elabore una aplicación en Java con interfaz gráfica que acepte desde la línea de comandos el número  $n$  de polígonos aleatorios a generarse. Al ejecutarse deberá crear una `List` de

polígonos, cada uno con un tamaño aleatorio, y con un numero de lados aleatorio menor a quince. Posteriormente deberá dibujar los n polígonos en posiciones aleatorias de la pantalla. El radio máximo de un polígono no deberá superar a 1/8 del alto de la pantalla.

Después de dibujar todos los polígonos deberá esperar tres segundos para que comience a desplazarse el polígono más grande en una dirección aleatoria y a una velocidad directamente proporcional a su tamaño (entre mas área debe ir mas lento y a menos área más rápido). Después de que el polígono más grande comience a desplazarse a una velocidad constante, deberá esperar un segundo para que comience el segundo polígono más grande a desplazarse en dirección aleatoria y así sucesivamente esperar un segundo entre el desplazamiento de cada polígono hasta terminar con todos los polígonos.

Para los gráficos utilice el siguiente código mínimo para dibujar un polígono de tres lados en una ventana. Favor de no utilizar otro tipo de bibliotecas debido a que se me complica la compilación de su código en mi computadora.

```
import javax.swing.*;
import java.awt.*;

public class SimpleGui2 extends JFrame
{
    public static void main(String[] args){
        SimpleGui2 gui = new SimpleGui2();
        gui.setVisible(true);
    }

    public SimpleGui2()
    {
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Panel p = new Panel();
        add(p);
    }

    private class Panel extends JPanel {
        @Override
        public void paintComponent(Graphics g){
            g.setColor(Color.blue);

            Polygon poligono=new Polygon();
            poligono.addPoint(0, 0);
            poligono.addPoint(100, 0);
            poligono.addPoint(50, 100);
            g.drawPolygon(poligono);
        }
    }
}
```

**Importante: Este proyecto es individual. Suba cada clase en un archivo separado, y cada archivo de código que suba debe contener al inicio como comentario el número de proyecto, su nombre completo y el grupo al que pertenece, de no hacerlo así se le descontará un punto de la calificación. No suba archivos comprimidos ni ligas a sitios web externos pues no le será tomado en cuenta el proyecto. Asimismo, deberá subir un video breve mostrando como se ejecuta su proyecto y que efectivamente realiza lo que se pide. Se recomienda utilizar OBS Studio con baja resolución y no es necesario que hable en el video.**