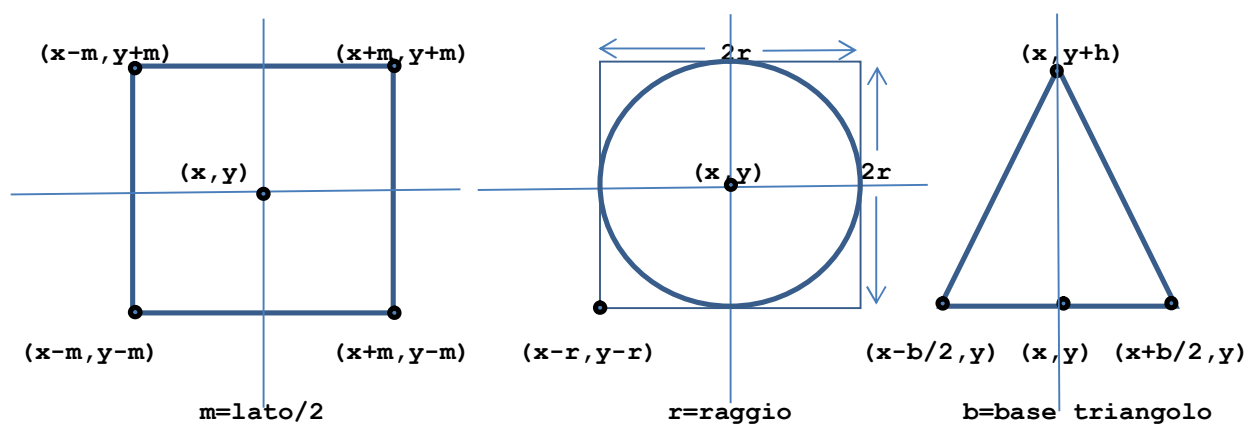


## Esercitazione 03

### Array di figure

Riprendete il disegno di quadrati e cerchi in un **Jframe** (in una finestra Java) visto nella Lezione 13. Ricopiate tutto il codice, quindi modificate le classi Quadrato e Cerchio aggiungendo attributi privati per le coordinate (interi)  $x, y$  del **centro** del quadrato e del centro del cerchio, e per un **colore**  $c$  (cercate informazioni sulla classe **Color** di Java). Aggiungete una classe Triangolo dei triangoli isosceli, descritti dalle coordinate  $x, y$  del **punto medio** della base, di base  $b$ , altezza  $h$  (tutti interi,  $h$  intero col segno), e colore  $c$ . Usate queste classi per costruire una finestra Java con disegnato un array contenente quadrati, cerchi, triangoli, di differente centro  $(x, y)$ , dimensione e colore  $c$ . I costruttori delle classi avranno quindi argomenti:

***Quadrato***  $(x, y, lato, c)$ , ***Cerchio***  $(x, y, r, c)$ , ***Triangolo***  $(x, y, b, h, c)$



**Nota.** La classe **Color** si trova nella libreria **java.awt**, e viene caricata dagli stessi comandi (visti nella Lezione 13) che caricano la classe **Jframe** delle finestre grafiche:

```
import java.awt.*;    //Abstract Window Toolkit (finestre grafiche)
import javax.swing.*; //estensione di awt per interfacce grafiche
```

Un colore si definisce con **Color.nome** (avete a disposizione i nomi: **black**, **red**, **green**, **yellow**, **blue** ...) oppure con **new Color(r,g,b)**, dove  $r, g, b$  sono interi da 0 a 255 che esprimono le proporzioni di rosso, verde e blu nel colore. Se cercate un colore su Wikipedia, trovate i valori di  $r, g, b$  necessari per definirlo.

## Esercitazione 03

### Estensione facoltativa dell'esercizio

Provate ad aggiungere un attributo privato **Color d** per Cerchi, Quadrati e Triangoli, per decidere il colore dell'interno della figura. Per disegnare l'interno di un poligono oppure di un ovale, e il poligono stesso, usate i seguenti comandi per oggetti della classe Graphics:

```
fillOval(int x, int y, int width, int height)  
fillPolygon(int[] xPoints, int[] yPoints, int nPoints)  
drawOval(int x, int y, int width, int height)  
drawPolygon(int[] xPoints, int[] yPoints, int nPoints)
```

Trovate le spiegazioni su internet. Per riempire una figura con il colore d, dovete prima usare il comando **g.setColor(d)** e dopo il comando fillOval, fillPolygon, esattamente come quando decidete il colore di una linea.

Non includiamo qui la soluzione della versione facoltativa dell'Esercitazione 03.