Object-Oriented Programs

Consist of

1. A static part (Class coding), that include
   - Identification of objects and their relationships
   - Developing a class diagram
   - Coding of the class diagram
2. A dynamic part (the code in main()), that consists of
   - Instantiation of objects in the main() program
   - Scheduling their interactions to implement the desired functionality in an organized way

Overview

- Identifying Objects
  - Identity
  - Properties
  - Behavior
- Identify Classes
  - Name
  - Class Attributes
  - Class Methods
- Identify Class Relationships (Hierarchies)
- Develop a class diagrams

Identifying Objects

Problem #1: 2D Geometric Objects

Regular 2D shapes can be polygons or circles. A polygon may be a triangle, a rectangle, square or a circle. We can assign a color to a shape and draw it. The shape can be moved to a position. We should be able to determine perimeter and area of a given shape.

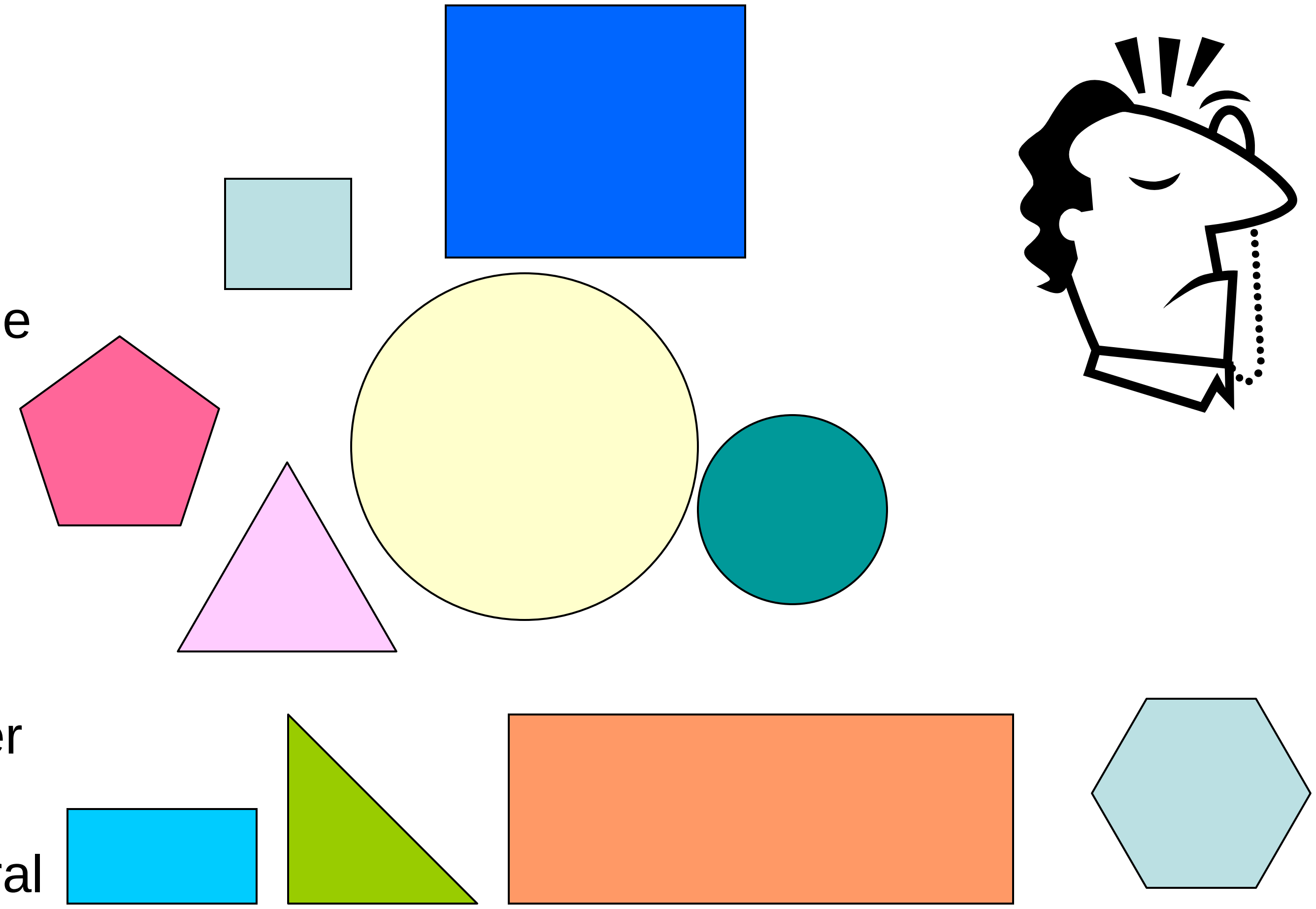
What do you notice?

Problem #1: 2D Geometric Shapes

Regular 2D shapes can be polygons or circles. A polygon consists of a number of points (>2). A polygon may be a triangle, a rectangle or a square. We can assign a color to a shape and draw it. The shape can be moved to a new position. We should be able determine perimeter and area of a given shape. In case of a triangle, we determine whether, it is equilateral or isosceles triangle.

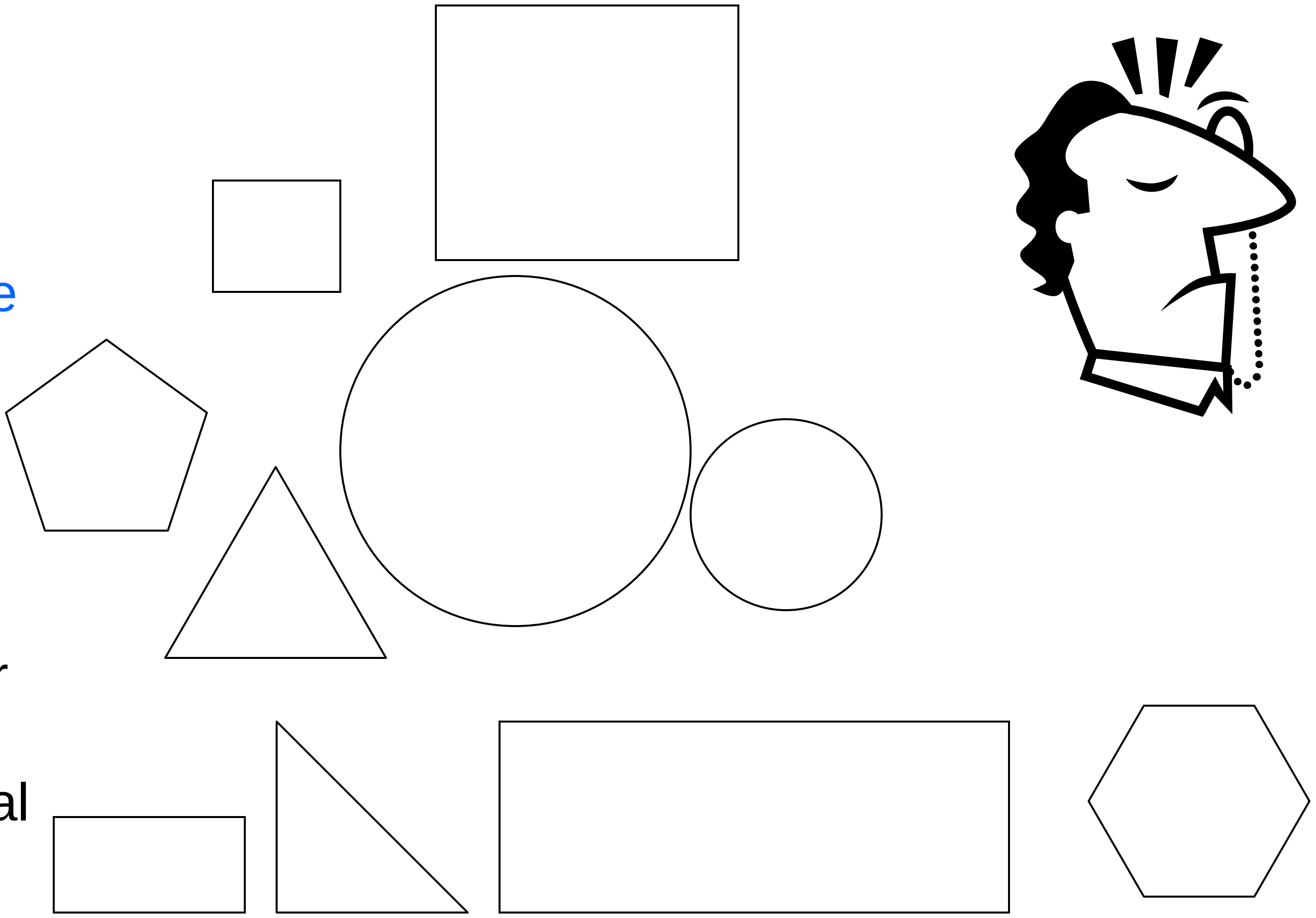
What do you notice?

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



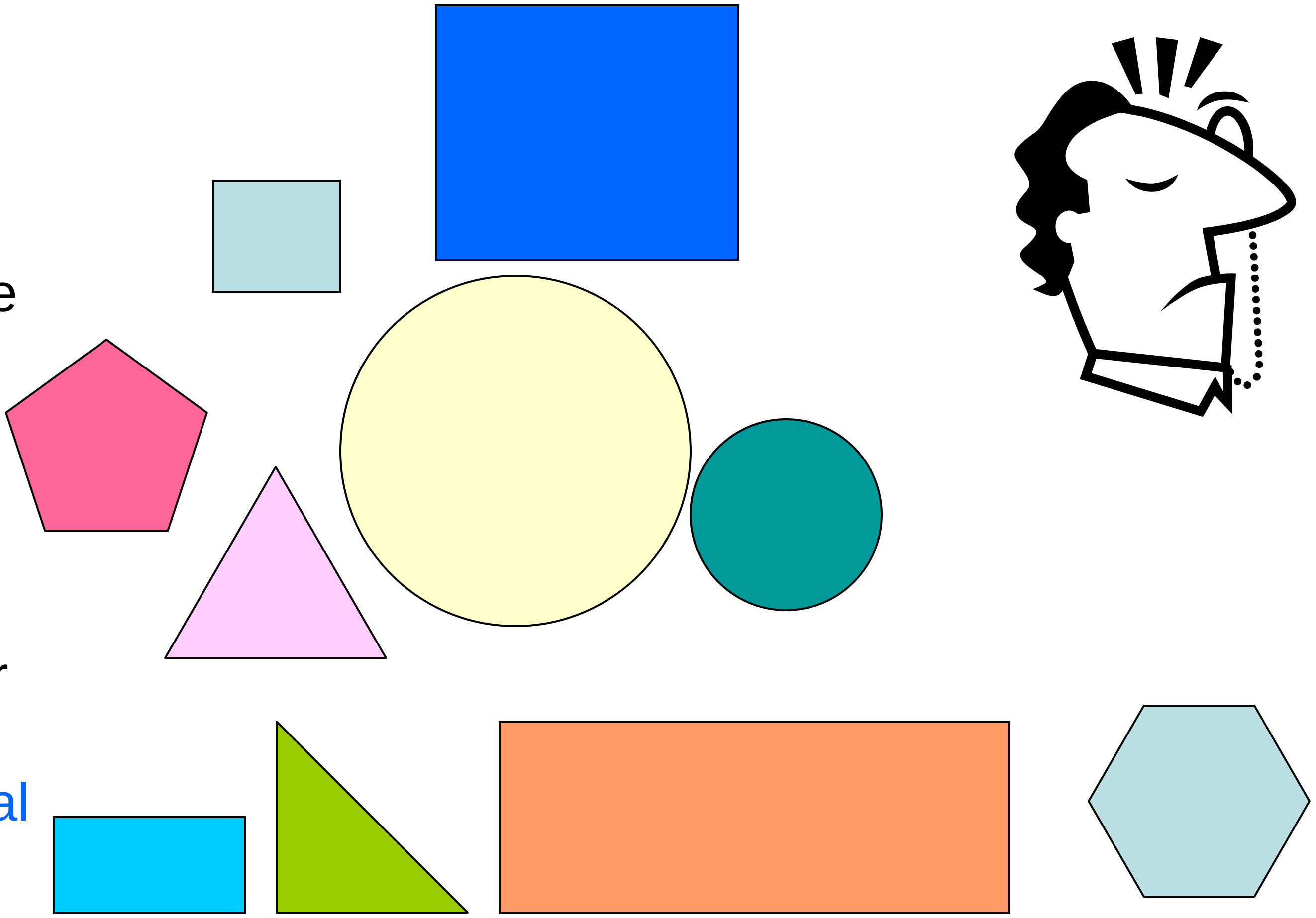
What do you notice? Objects!

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



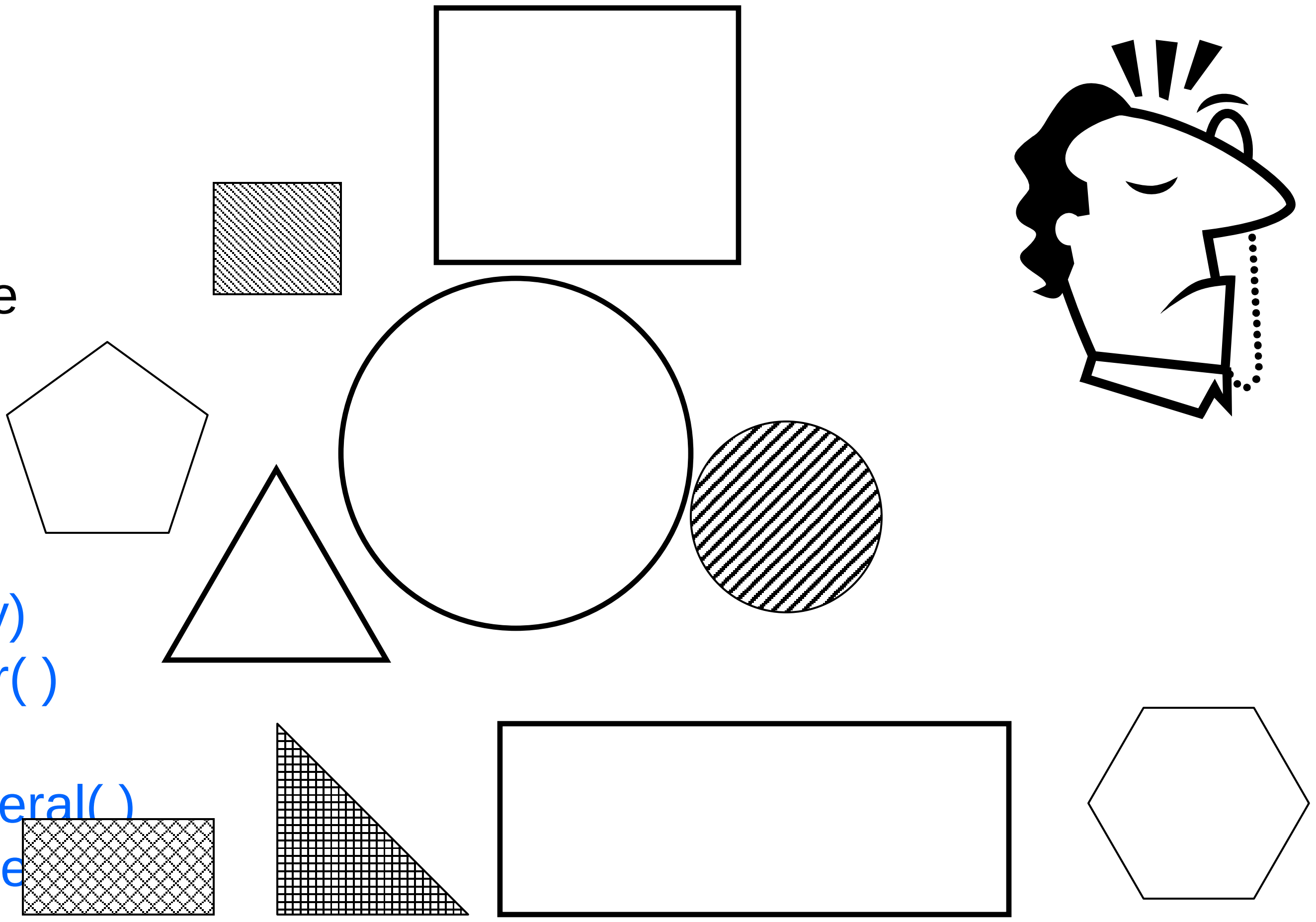
What do you notice? Properties!

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



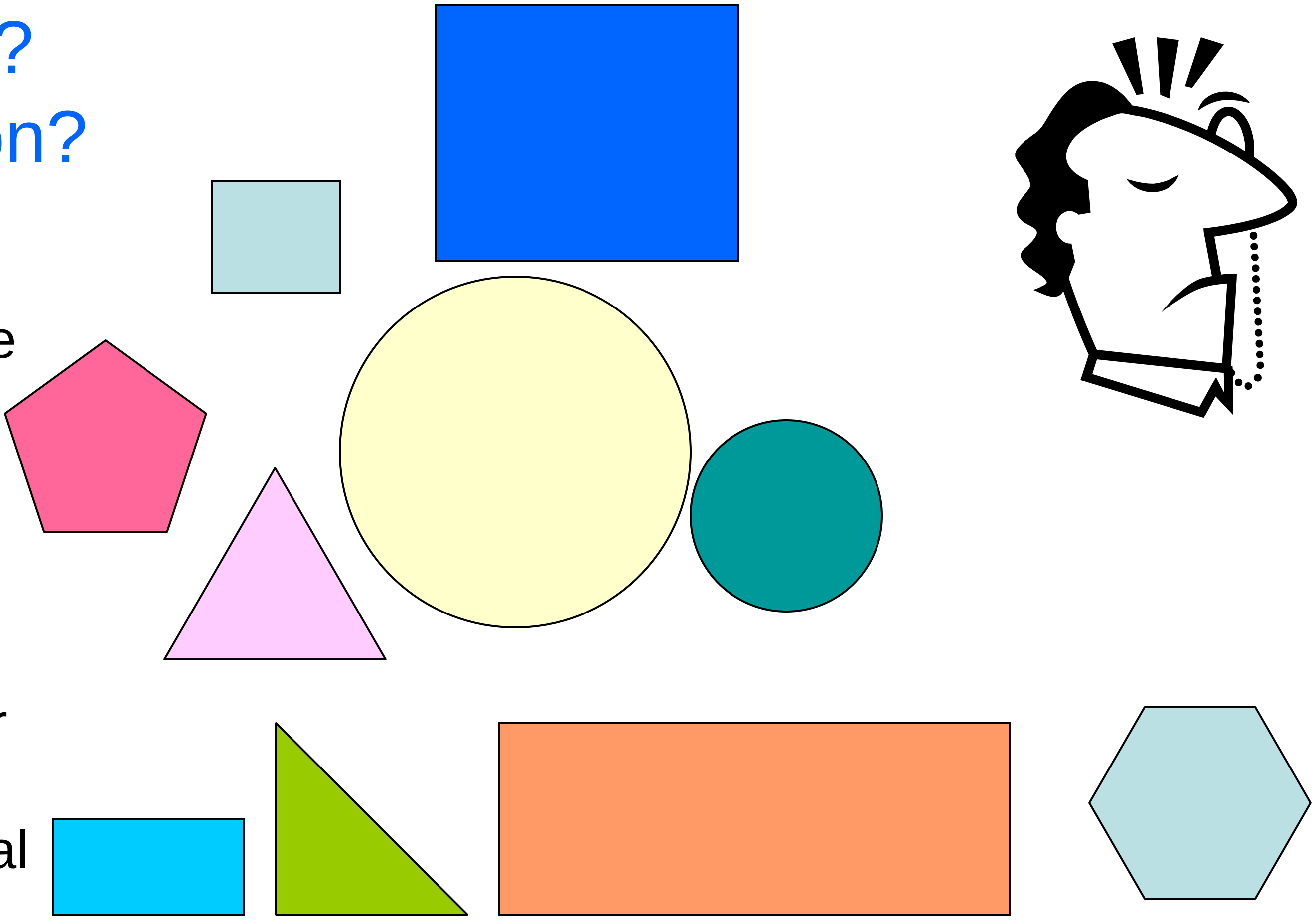
What do you notice? Behavior!

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw( )
- Move(x, y)
- Perimeter( )
- Area( )
- Is Equilateral( )
- Is Isoscele



And what else?

- Shape?
- Polygon?
- Circle
- Triangle
- Rectangle
- Square
- Color
- Point
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles

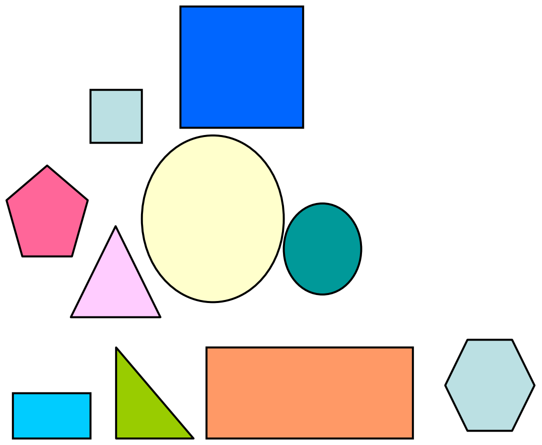


And what else?

- Shape?
- Polygon?
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles

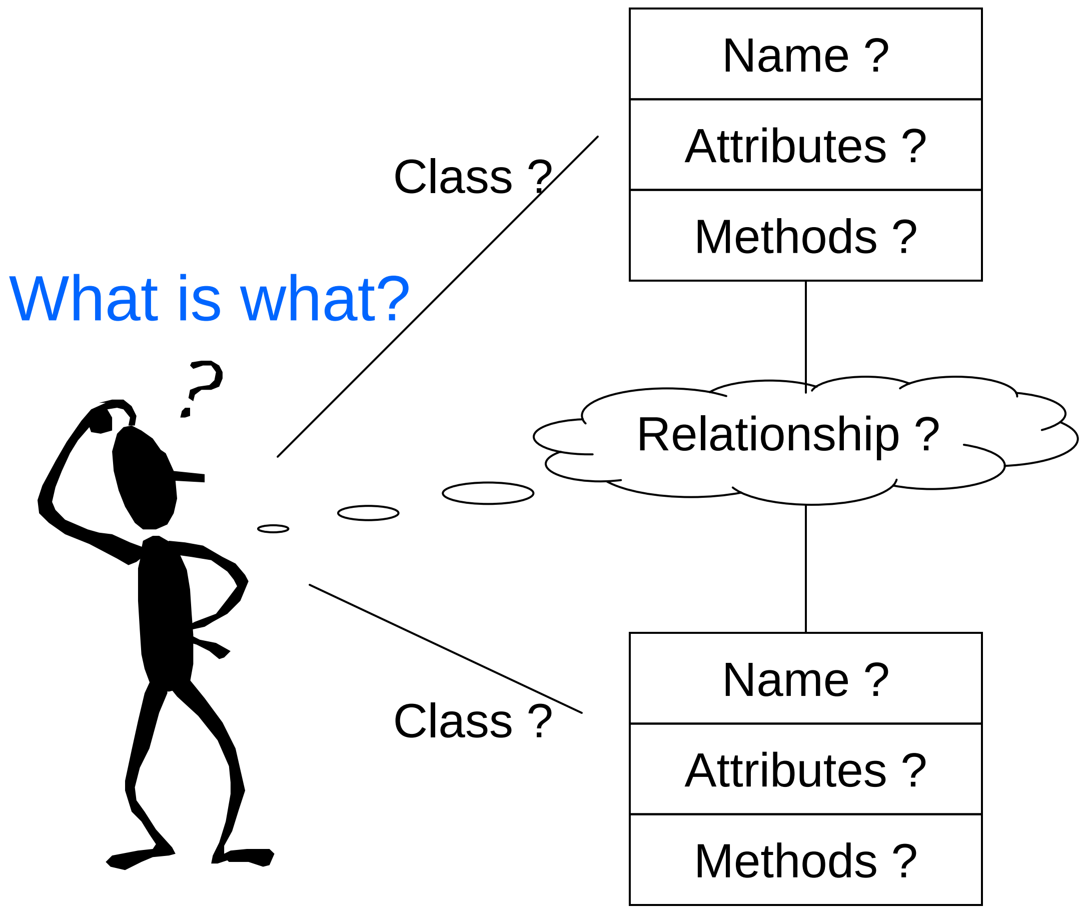
- All are Shapes !
- Some of them are Polygons !
- A "kind-of" relationship
- Both are abstract !

- A Polygons consists of a number of points
- A "has-a" or "part-of" relationship!



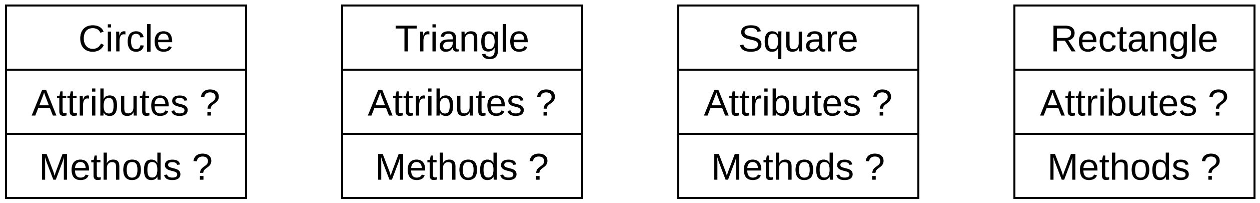
Classes?

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



Classes

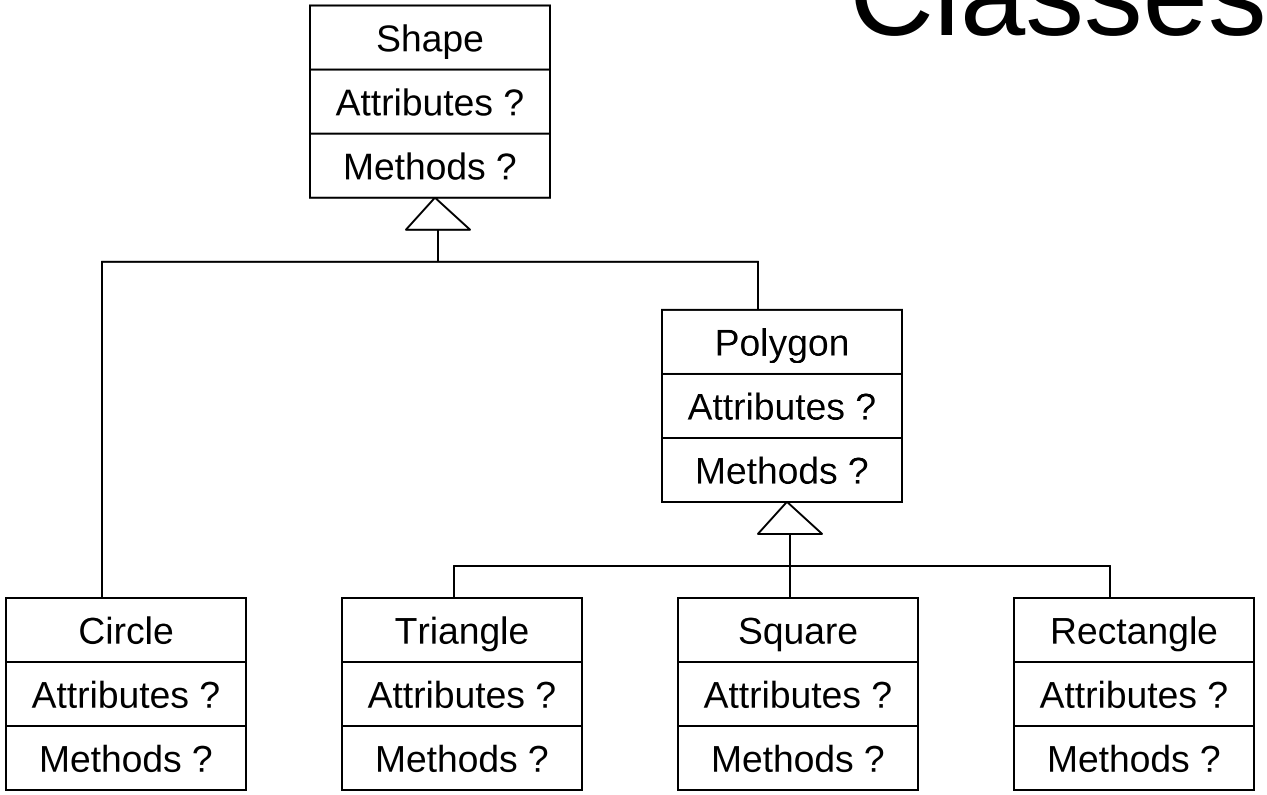
- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



One class per different type of objects

Classes

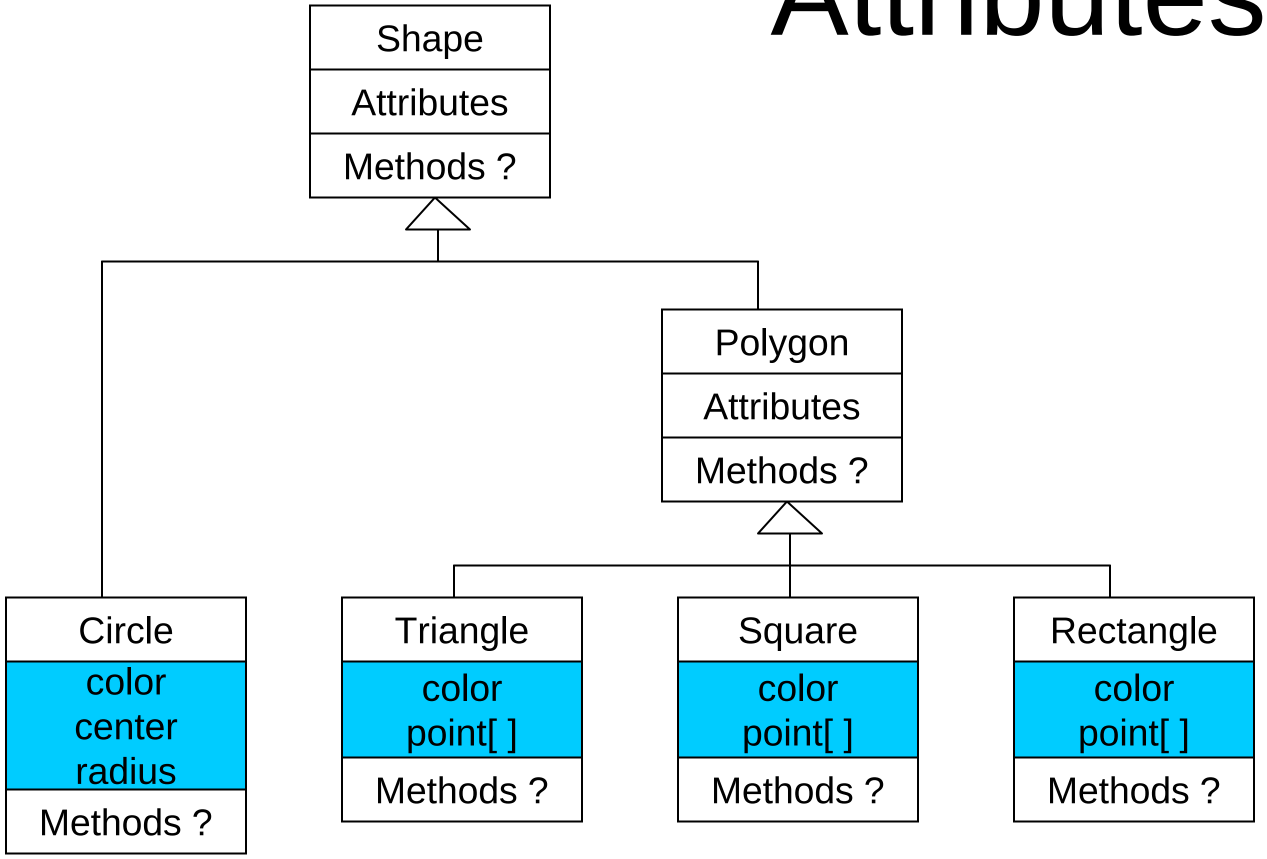
- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles



One class per different type of objects

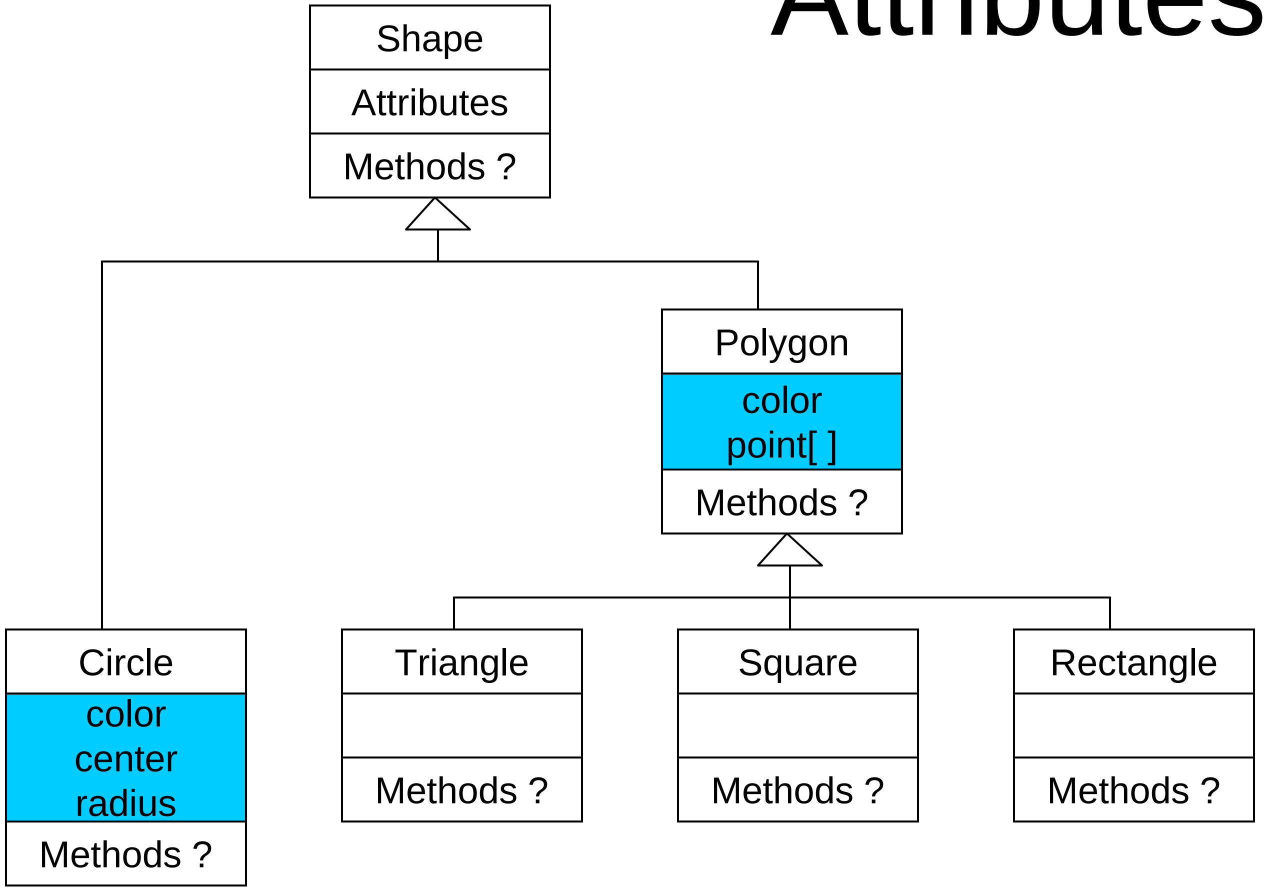
Attributes

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles
- getColor
- setColor



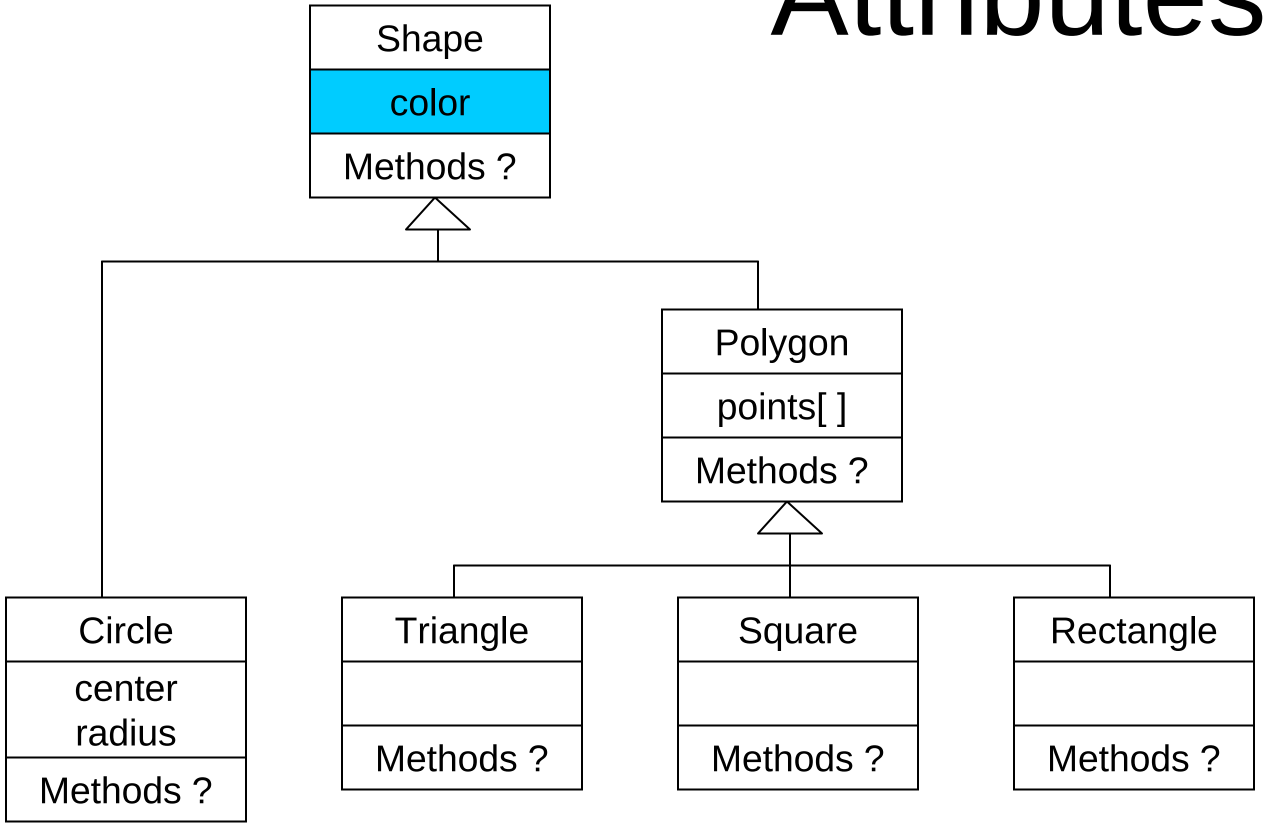
Attributes

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles
- getColor
- setColor



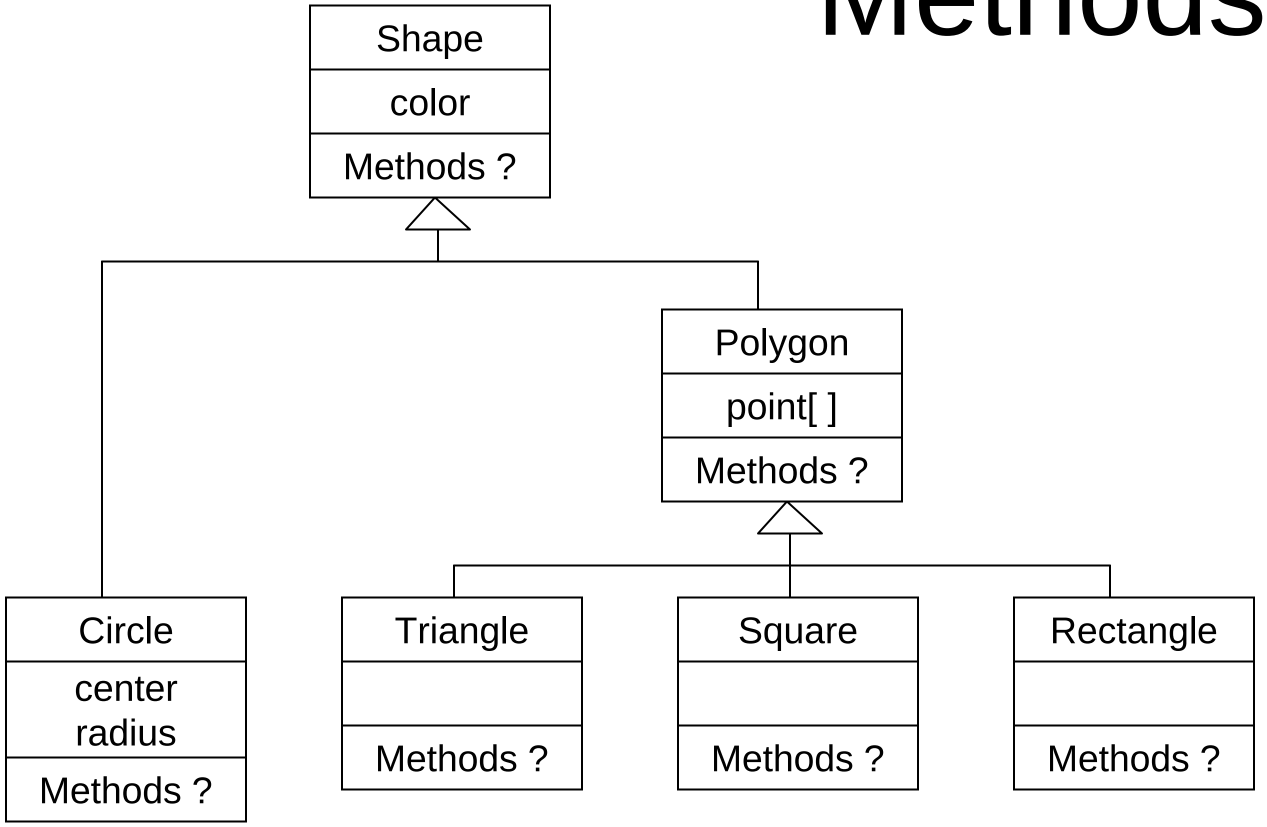
Attributes

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles
- getColor
- setColor



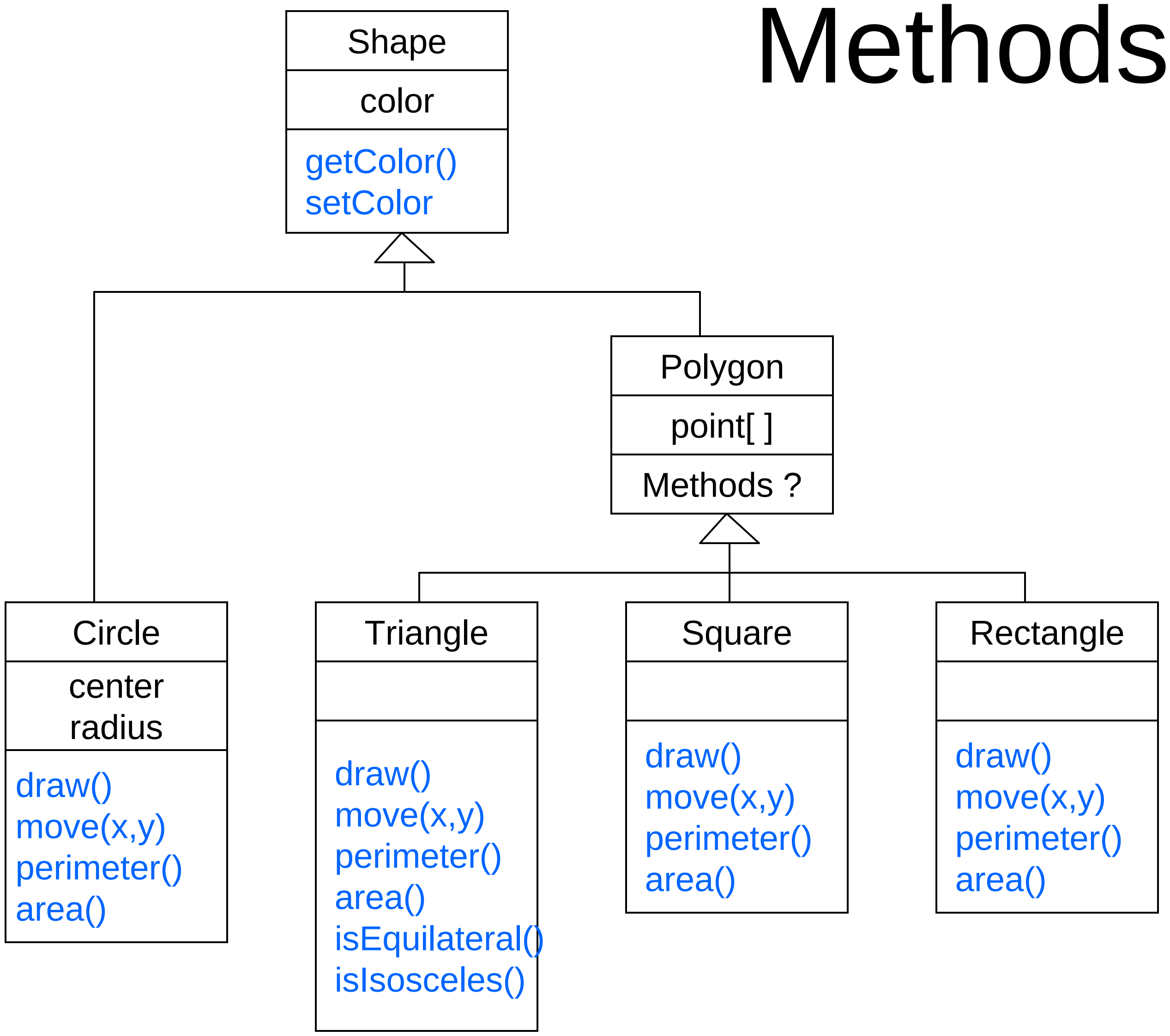
Methods

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- Draw
- Move
- Perimeter
- Area
- Equilateral
- Isosceles
- getColor
- setColor



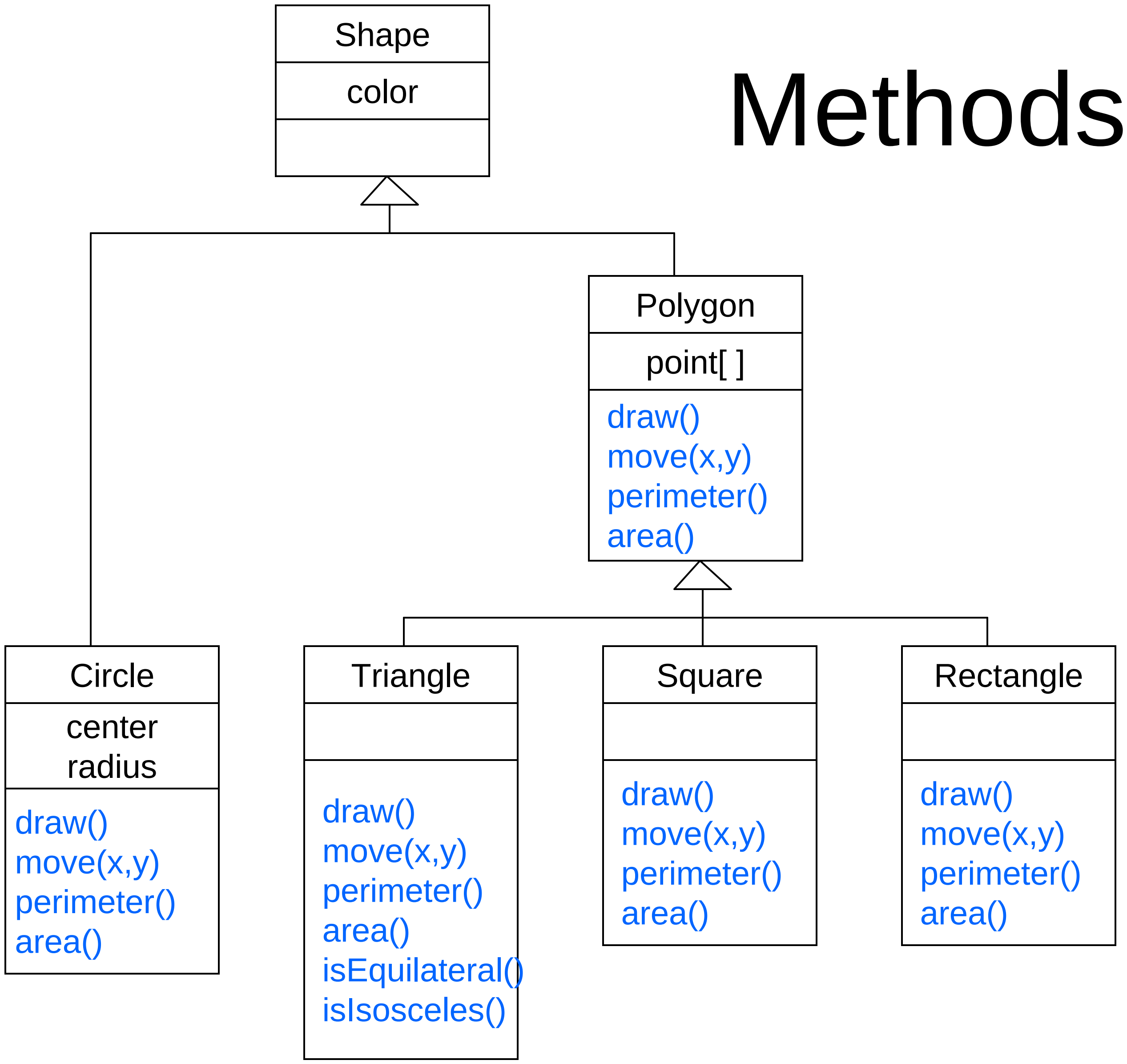
Methods

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- draw()
- move()
- perimeter()
- area()
- isEquilateral()
- isIsosceles()
- getColor()
- setColor()



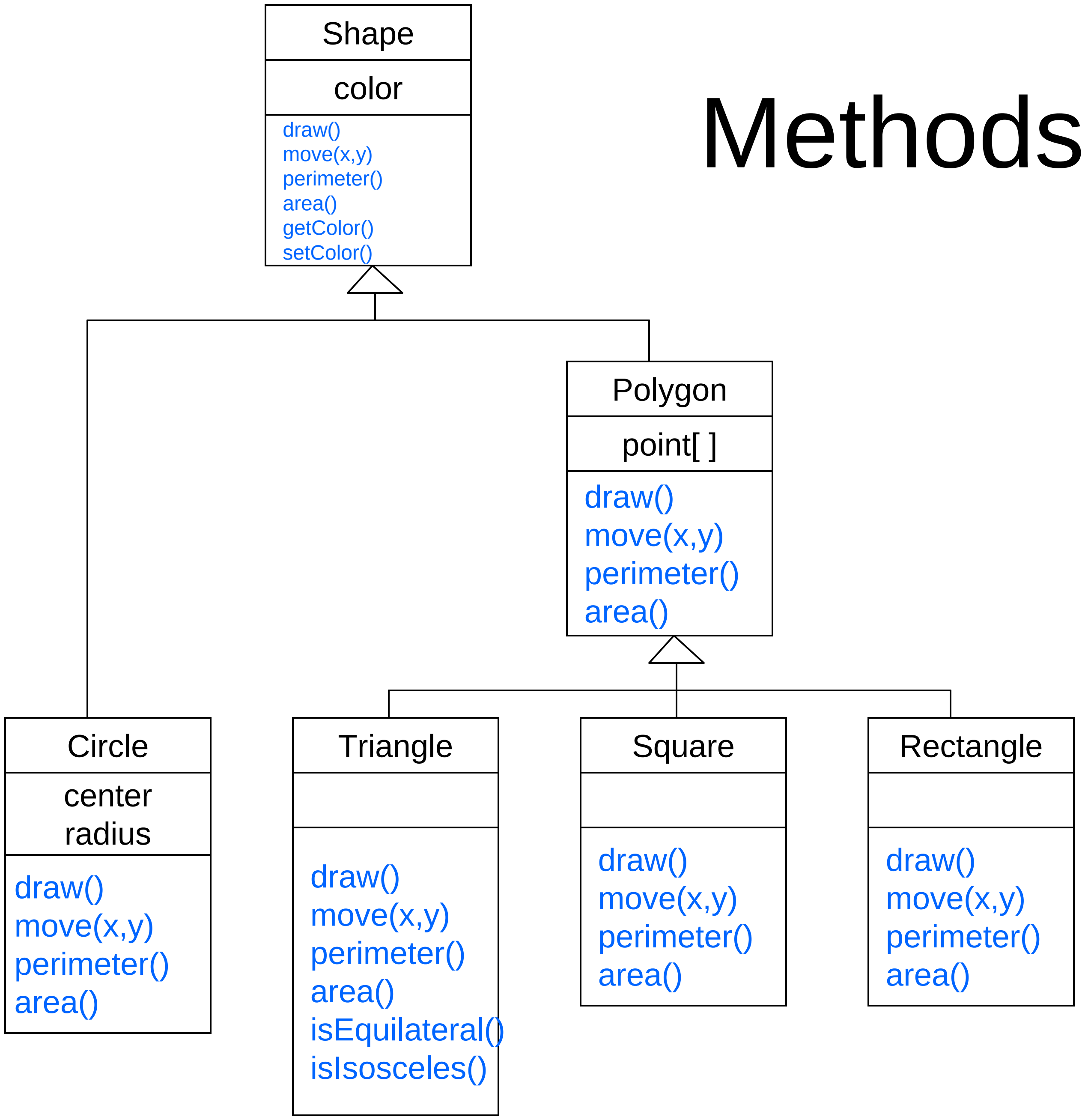
Methods

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- draw()
- move()
- perimeter()
- area()
- isEquilateral()
- isIsosceles()



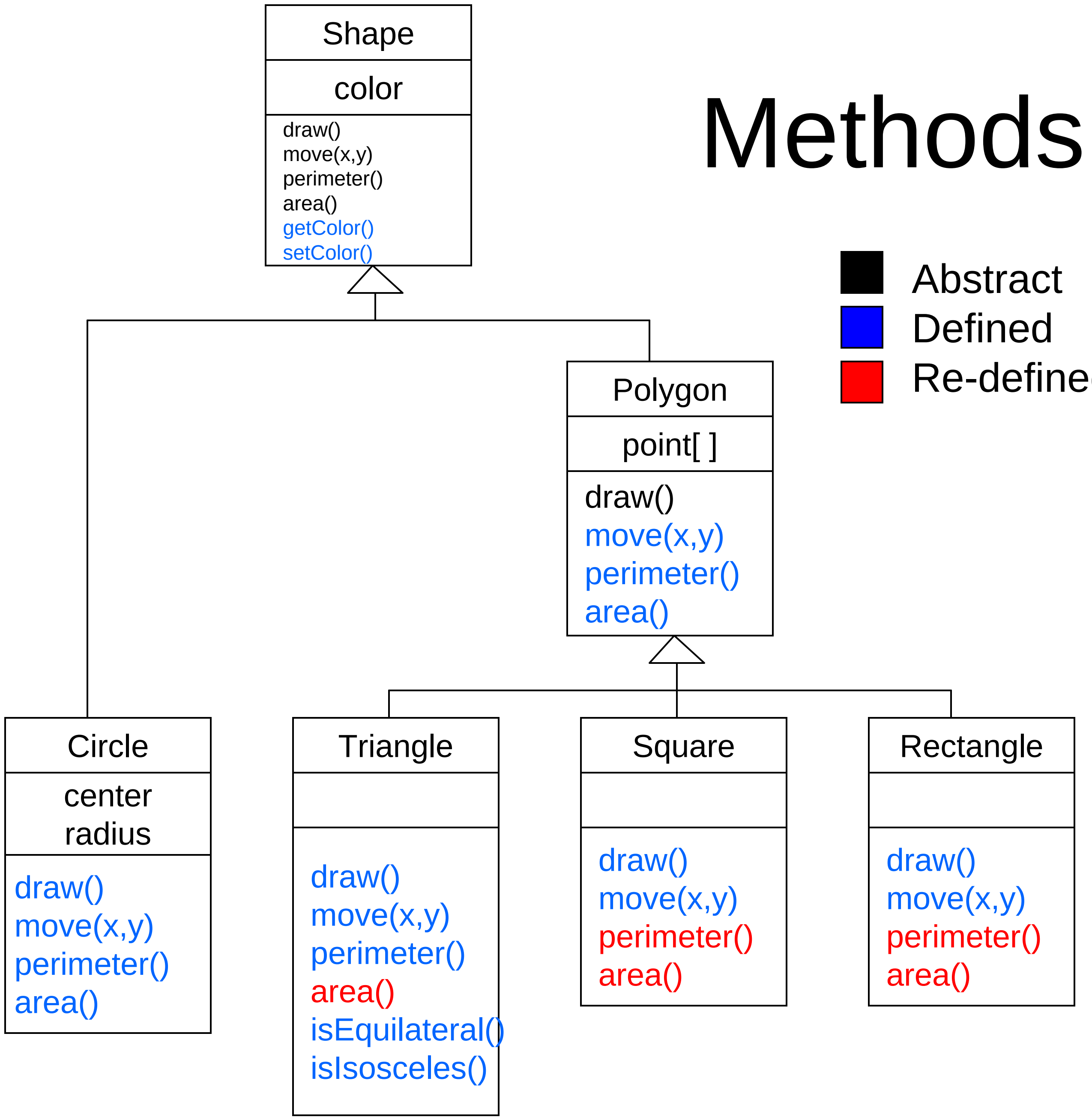
Methods

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- draw()
- move()
- perimeter()
- area()
- isEquilateral()
- isIsosceles()



Methods

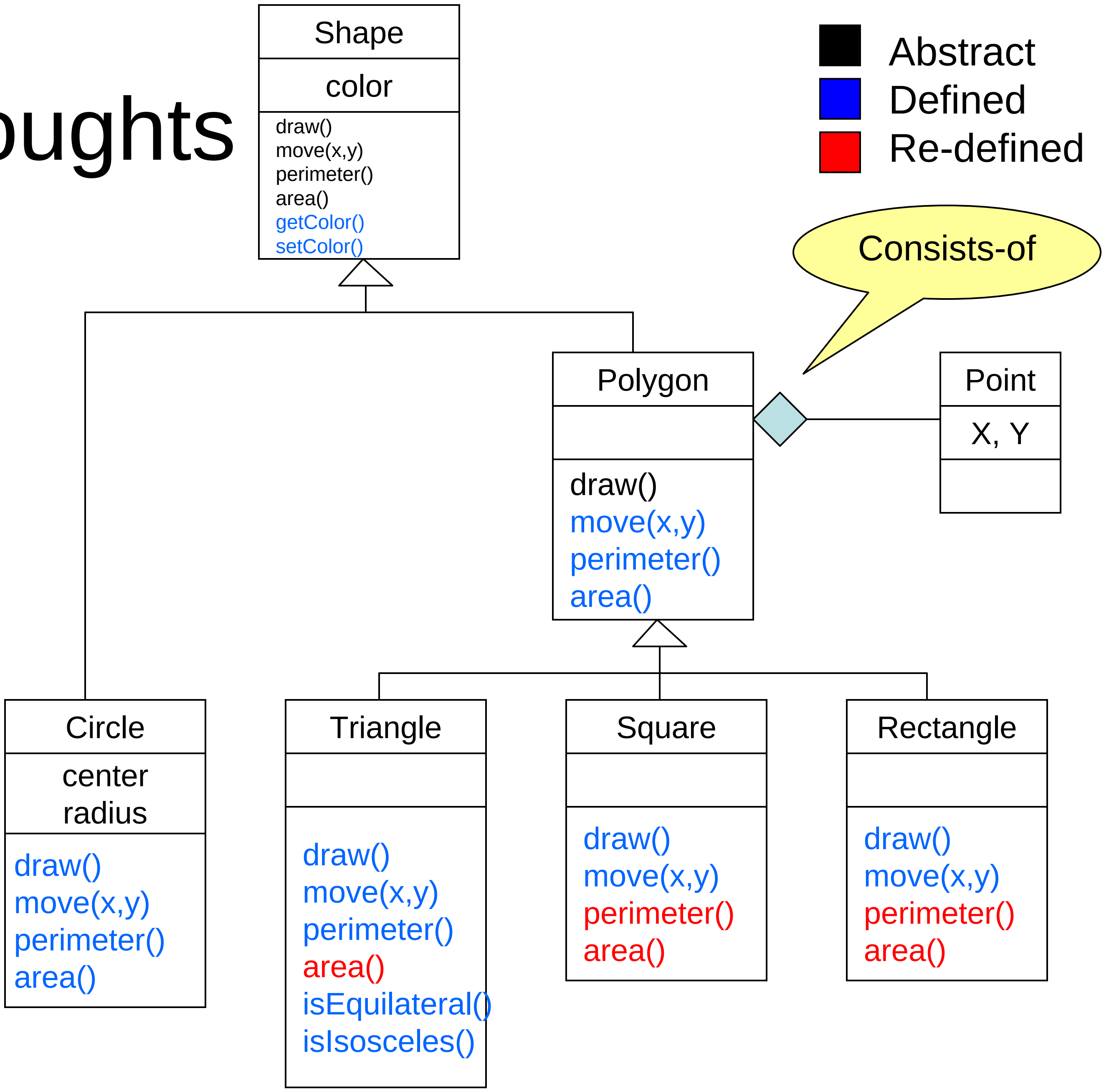
- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- draw()
- move()
- perimeter()
- area()
- isEquilateral()
- isIsosceles()



⬛	Abstract
🟦	Defined
🟥	Re-defined

Final Thoughts

- Shape
- Polygon
- Circle
- Triangle
- Rectangle
- Square
- Point
- Color
- draw()
- move()
- perimeter()
- area()
- isEquilateral()
- isIsosceles()



Summary

- Identify objects in a given problem
  - Identify Identity
  - Identify properties
  - Identify behavior
- Identify relationships between objects
- Correspondingly develop a class diagram