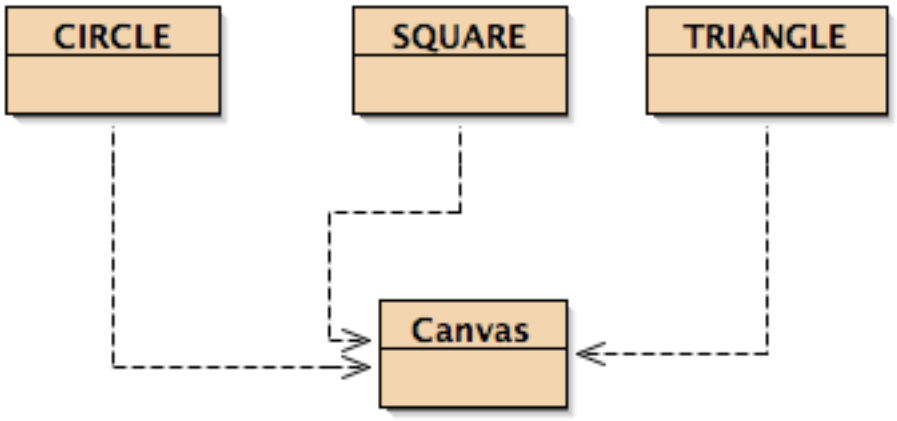
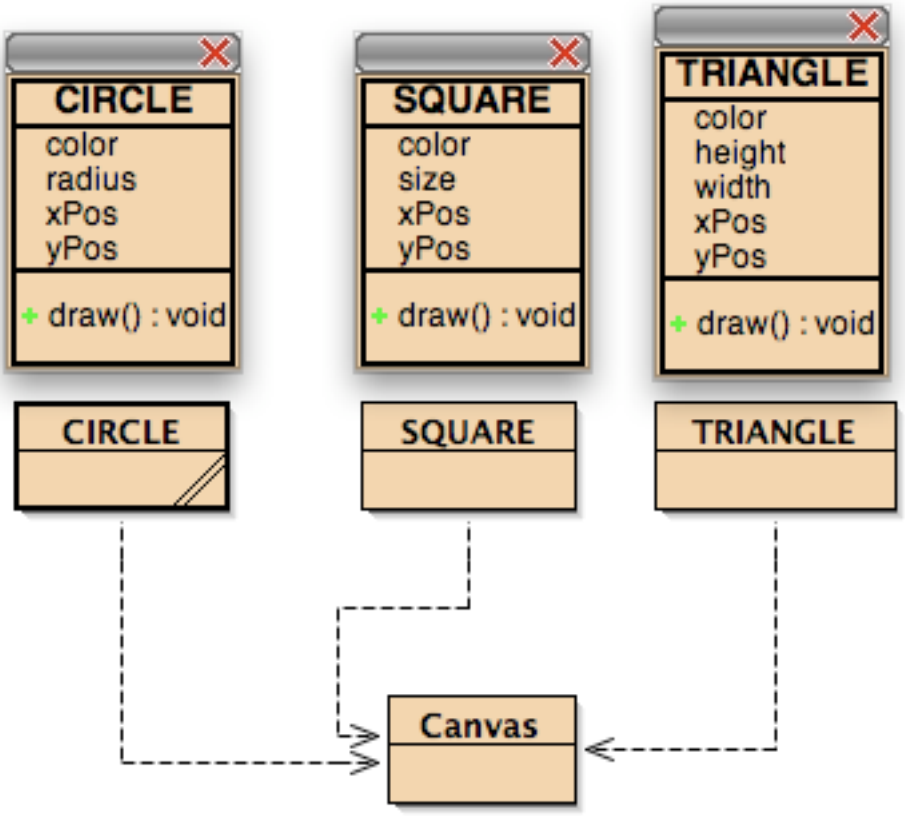
Vererbung

Die drei Klassen CIRCLE, SQUARE und TRIANGLE haben bislang nichts miteinander zu tun. Sie können sich zwar selbst in ein und dasselbe Canvas (engl. Leinwand) zeichnen. Mehr aber auch nicht.



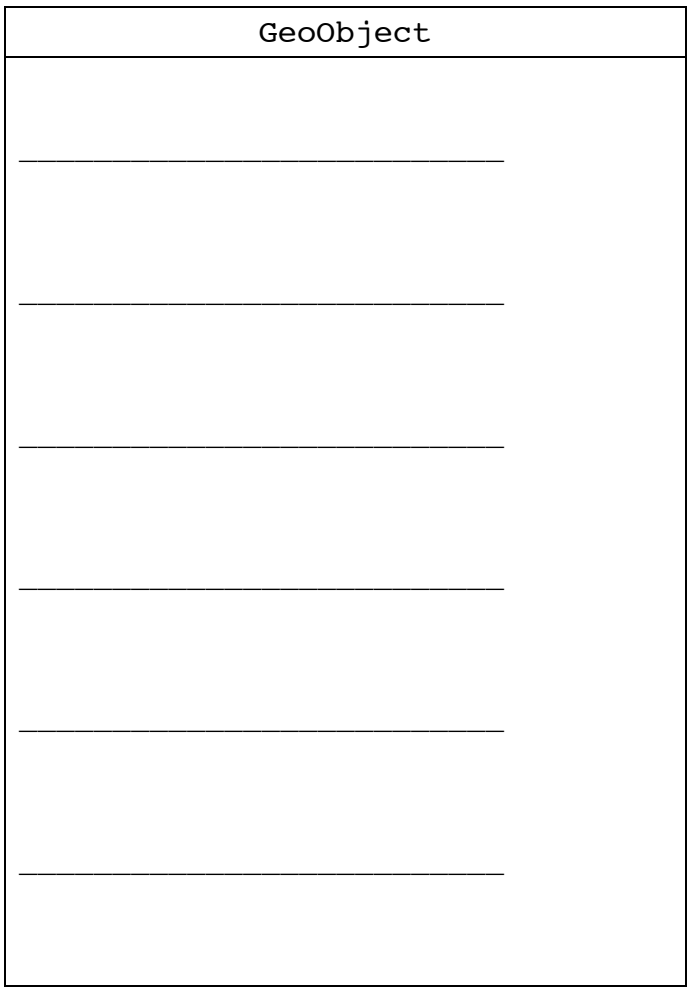
Schaut man sich aber die Klassenkarten genauer an, so erkennt man Gemeinsamkeiten:



Welche Attribute und Methoden haben die drei Klassen gemeinsam?

Es ist daher sinnvoll zu sagen, dass Klassen als Basis eine gemeinsame Klasse haben: `GeoObject`.

Wie müsste das erweiterte Klassendiagramm dieser Klasse `GeoObject` aussehen?



Damit die Klassen CIRCLE, SQUARE und TRIANGLE diese neue Klasse als Basis nehmen muss man den Kopf der Klassen-Definition wie folgt erweitern:

```
public class CIRCLE _____________________________
```

Ebenso kann der Konstruktor der Klasse CIRCLE angepasst werden:

```
public CIRCLE() {

    ____________________________________________
    radius = 30;
}
```

Musterlösung

```
class GeoObject {
    //Die Attribute werden auf protected gesetzt,
    //damit die abgeleiteten Kindklassen, aber auch nur diese,
    //auf die Attribute zugreifen können.

    protected int xPos;
    protected int yPos;
    protected String color;

    // Konstruktoren
    public GeoObject() {
        // Der Standardkonstruktor wird immer benoetigt
        xPos = 10;
        yPos = 10;
        color = "black";
    }

    public GeoObject(int neueXPos, int neueYPos, String neueFarbe){
        xPos = neueXPos;
        yPos = neueYPos;
        color = neueFarbe;
    }

    // Dienste
    // Jedes Objekt muss sich zeichnen können, aber jedes wird
anders gezeichnet.
    public void draw() {}
}
```

```
import java.awt.*;
import java.awt.geom.*;

public class CIRCLE extends GeoObject{
    //xPos, yPos, color sind in der Elternklasse GeoObject definiert
    private int radius;

    public CIRCLE() {
        //Es muss immer erst der Konstruktor der Elternklasse
        //GeoObject aufgerufen werden
        super(20, 60, "blue");
        radius = 30;
    }

    public CIRCLE(int neuesR, int newX, int newY, String newColor) {
        super(newX, newY, newColor);
        radius = neuesR;
        draw();
    }

    public void draw() {
        Canvas canvas = Canvas.getCanvas();
        canvas.draw(this, color, new Ellipse2D.Double(xPos, yPos,
radius, radius));
    }
}
```