

Thema: Objektorientiertes Entwerfen / Modellieren / Implementieren mit Java

Als Ausgangspunkt gegeben ist die Klasse **DrawingBoard**, in der ein Fenster als Zeichenfläche sowie ein Stift angelegt werden :

```
public class DrawingBoard
{
    Stift      stift;      // Objektvariablen -> Hat-Beziehung
    Bildschirm fenster;

    //----- Konstruktor -----
    public DrawingBoard()
    {
        fenster = new Bildschirm(600, 600);
        stift = new Stift();
    }

    //----- Dienste -----
    public void loesche_alles()
    {
        fenster.loescheAlles();
    }

    public void programm_beenden()
    {
        System.exit(0);
    }
} // class
```

Darüber hinaus werden der Dienst **loesche_alles()**, der den gesamten aktuellen Fensterinhalt löscht, sowie der Dienst **programm_beenden()** zur Verfügung gestellt.

Arbeitsauftrag 1:

Entwerfe das *UML*-Klassendiagramm einer Klasse **PolyForms**, die von der Klasse **DrawingBoard** abgeleitet wird (-> Spezialisierung / Ist-Beziehung).

Sie soll zunächst nur die beiden parameterlosen öffentlichen Dienste **zeichne()** und **loesche()** für das Zeichnen und Löschen unterschiedlicher grafischer Figuren zur Verfügung stellen (neben den *set-* und *get-*Methoden, s.u.). Weitere Dienste werden später im Entwurfsprozess detailliert.

Allen graphischen Figuren gemeinsam ist die Charakterisierung durch die fünf Eigenschaften **x-** und **y-**Position auf der Zeichenfläche, **Größe** in Pixel, **Winkel** gegenüber der Horizontalen (karthesisches Koordinatensystem), sowie eine ganzzahlige **Kennnummer**, die die jeweilige graphische Form bezeichnet (Wertebereich 1 – 5).

Für alle Eigenschaften einer grafischen Figur sollen **set-** und **get-Methoden** zur Verfügung stehen, mit denen Werte gesetzt werden (-> Aufträge) und Werte abgefragt werden (-> Anfragen) können.

Darüber hinaus soll die Klasse drei unterschiedliche Konstruktoren bieten, die bei der Erzeugung einer Instanz der Klasse

- 1) die Vorgabe der Werte aller Eigenschaften
- 2) die Vorgabe nur der Kennnummer der Figur sowie
- 3) eine parameterlose Instanziierung ohne weitere Wertvorgabe

ermöglichen.