

**Thema:** Objektorientiertes Entwerfen / Modellieren / Implementieren mit Java

Als Ausgangspunkt gegeben ist der *UML*-Entwurf der Klasse *PolyForms*, sowie dessen Umsetzung in *Java*-Quellcode gemäß Arbeitsblatt 2.

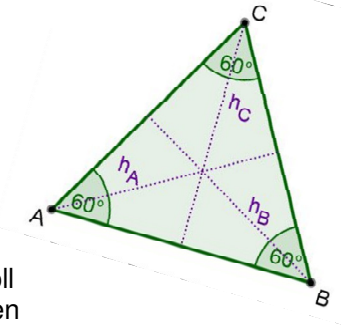
Im folgenden werden wir verschiedene Methoden ergänzen, um unterschiedliche geometrische Figuren im Ausgabefenster zu zeichnen und damit die Fähigkeiten der Klasse erweitern.

### Arbeitsauftrag 3:

Die Klasse *PolyForms* soll um den privaten Dienst

**zeichneDreieck()**

erweitert werden. Ergänze das *UML*-Diagramm der Klasse entsprechend.



#### Teil a):

Entwerfe in Pseudocode die private Methode **zeichneDreieck()**. Diese soll ein gleichseitiges Dreieck auf der Basis der Klassenattribute / Eigenschaften der Klasse *PolyForms* zeichnen.

Der Bezugspunkt für die Position des Dreiecks soll der Schwerpunkt des Dreiecks sein (ist gleichzeitig Höhenschnittpunkt und Schnittpunkt der Seiten- und Winkelhalbierenden sowie Mittelpunkt von In- und Umkreis des Dreiecks, siehe Skizze). Das Attribut **size** gibt die Seitenlänge vor, das Attribut **angle** den Winkel gegenüber der Horizontalen.

Die Kennnummer **shape** für Dreiecke ist 1.

Sollten Fragen zu mathematischen Grundlagen in Verbindung mit Dreiecken auftreten, formuliere diese im Plenum oder kläre sie mit Hilfe der Mathematik-Formelsammlung.

**Hinweis:** Eine Übersicht über die Verwendung von Pseudocode ist im Kursbereich meiner Seite ([www.milzners.de/kursbereich](http://www.milzners.de/kursbereich)) zu finden.

#### Teil b):

Übersetze, nach Besprechung des Pseudocode-Entwurfes aus a), dann den Entwurf in syntaktisch korrekten und vollständigen *Java*-Quellcode (auf Papier!).

**Hinweis:** Sollten mathematische Funktionen oder Konstanten für die Berechnung erforderlich sein, sind diese meist in der *Java*-Basisklasse **Math** zu finden (z.B. `Math.PI` oder `Math.sqrt(...)` für Quadratwurzeln).

#### Teil c):

Nach der Diskussion / Vorstellung im Plenum implementiere die Methode als weiteren Dienst der Klasse **PolyForms**.

Ergänze dann die Methode **zeichne()** um einen geeigneten Aufruf der Methode **zeichneDreieck()**, damit diese aus dem Kontextmenü einer Objektinstanz heraus in *BlueJ* aufgerufen werden kann (siehe Arbeitsblatt 2).

Erprobe ihre Korrektheit durch das Erzeugen verschiedener Ausprägungen von Dreiecken (Größe, Position, Winkel) mit Hilfe der *Set*-Methoden sowie der *zeichne()* und *loesche()*-Methode.