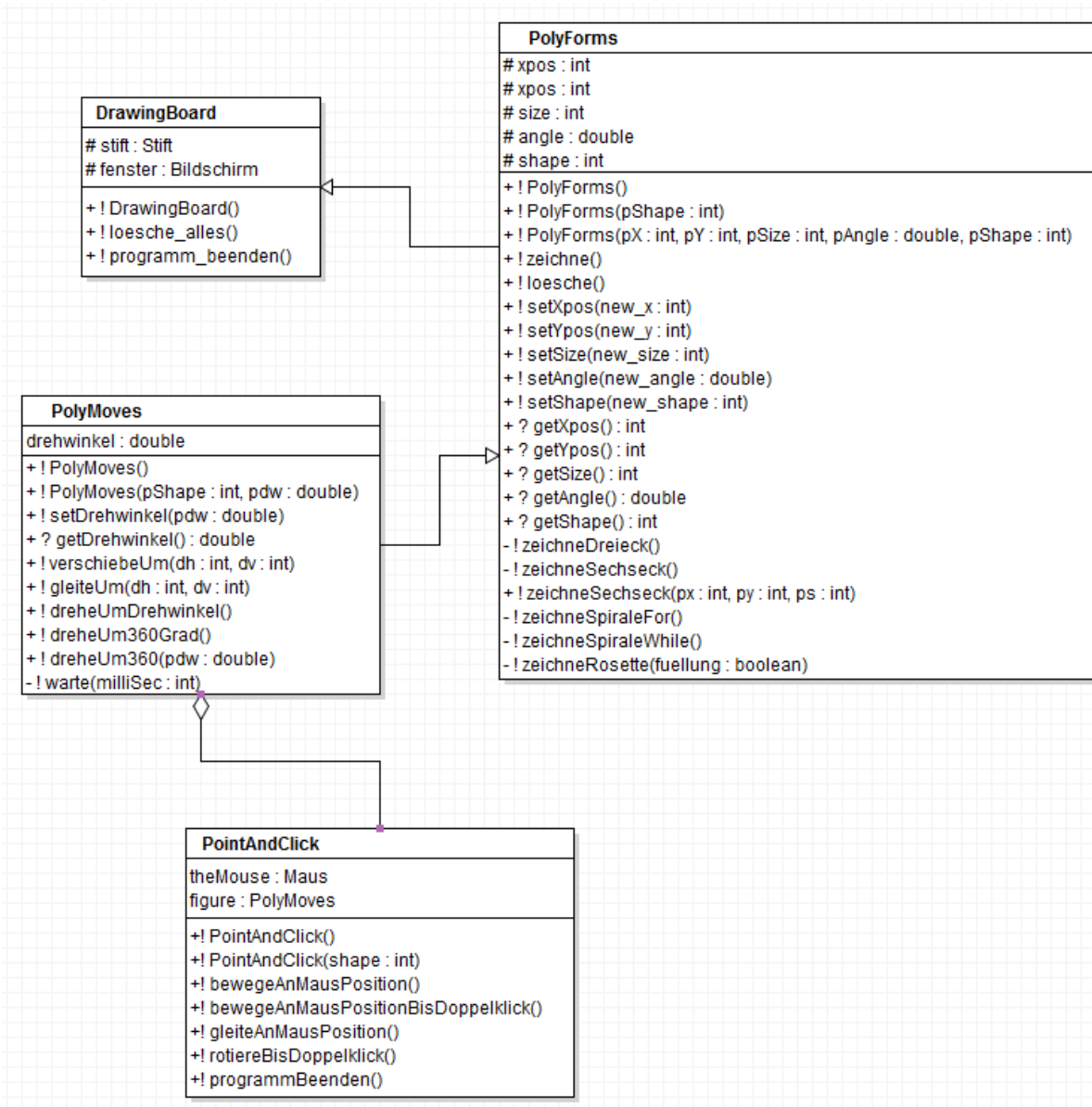


Thema: Objektorientiertes Entwerfen / Modellieren / Implementieren mit Java

Als Ausgangspunkt des Teilprojektes **PointAndClick** ist eine vollständige und lauffähige Implementierung der bisherigen Klassen *DrawingBoard*, *PolyForms* und *PolyMoves* erforderlich.



Alle Klassen sind in den Kursdateien der HPI-Cloud zum Download verfügbar, sodass in Verbindung mit den gegebenen Musterlösungen hier jeder den gleichen (Wieder-)Aufsetzpunkt im Projekt hat.

Nachdem mit den bisherigen Klassen das Prinzip der Vererbung / Spezialisierung (Ist-Beziehung) thematisiert wurde, steht jetzt eine Verwendungs-Beziehung (Hat-Beziehung) im Fokus.

Das obige **UML-Klassendiagramm** beschreibt die Struktur der Klasse **PointAndClick** vollständig.

Die Klasse *PointAndClick* verwendet die zuletzt erstellte Klasse *PolyMoves*. Sie stellt verschiedene Dienste zur Verfügung, um mit Hilfe einer *PolyMoves*-Instanz gezeichnete graphische Objekte an eine im Fenster mit der Maus angeklickte Position zu bewegen oder diese rotieren zu lassen.

Die verschiedenen Methoden der Klasse werden, wie bereits von den vorhergehenden Klassen bekannt, nach der Implementierung der grundlegenden Klassenstruktur gemäß UML-Diagramm inhaltlich konkretisiert.

Arbeitsauftrag 1:

Erzeuge in *BlueJ* die Klasse ***PointAndClick*** und übersetze hierin die in obigem Klassendiagramm dargestellte Klassenstruktur vollständig in *Java-Quellcode*.

Die beiden Konstruktoren der Klasse sollten *vollständig* implementiert sein. Alle weiteren Methoden müssen vorhanden sein, weisen aber zunächst nur einen leeren Methodenrumpf auf, der im Verlauf des Teilprojektes vervollständigt werden wird.

Wird in *BlueJ* eine Instanz der Klasse *PointAndClick* erzeugt, soll unmittelbar die gewählte graphische Struktur (also Dreieck / Sechseck / Rosette etc.) gezeichnet werden.

Da wir die Maus das erste Mal in einem Projekt verwenden, findet ihr der Einfachheit halber nachfolgend eine Übersicht der Methoden der Klasse *Maus* als Kopie der *SuM*-Onlinehilfe in *BlueJ*. Diese werden im weiteren Verlauf des Teilprojektes zum Einsatz kommen.

Constructor Summary	
Maus ()	Die Maus wird initialisiert.
Method Summary	
boolean doppelKlick ()	Prueft, ob ein Doppelklick stattgefunden hat.
void gibFrei ()	Dummy-Prozedur
int hPosition ()	liefert die gegenwaertige horizontale Koordinate der Position der Maus auf dem Bildschirm, unabhaengig davon, ob die Maus gedrueckt wurde.
boolean istGedrueckt ()	Prueft, ob eine Maustaste im Moment gedrueckt ist.
boolean spezialKlick ()	Aus Kompatibilitätsgründen: <code>spezialKlick()</code> heisst jetzt <code>doppelKlick()</code> .
int vPosition ()	liefert die gegenwaertige vertikale Koordinate der Position der Maus auf dem Bildschirm, unabhaengig davon, ob die Maus gedrueckt wurde.