

Anwendung von Artefakten in den Phasen des SE-Lebenszyklus (ST-Übungen)	Analyse (OOA)/ aUML	Design (OOD)/ dUML	Implementierung + Test (OOP)/ jUML <sup>1</sup>
Anwendungsfalldiagramm (UML)	x <sup>2</sup>		_3
Klassendiagramm (UML)	Domänenmodell/ Analyseklassendiagramm	Entwurfsklassendiagramm mit Kollaborationen für Entwurfsmuster	
Objektdiagramm (UML)	x		x
Zustandsdiagramm (UML) /Protokollmaschine (PSM) + Verhaltenszustandsmaschine (BSM) <sup>4</sup>	OLC, ULC, CSM		OLC, CSM
Aktivitätsdiagramm (UML)	Typischerweise für Workflows/ Geschäftsabläufe		-
Sequenzdiagramm (UML)	Typischerweise zur Modellierung von Szenarien für einen Anwendungsfall		Typischerweise zur Modellierung komplexer Methoden
Komponentendiagramm (UML)	Kontextdiagramm Top-Level-Architektur		Verfeinerte, komponentenbasierte Softwarearchitekturen
Blockdiagramm			-
CRC-Karten	x (verhaltensgetriebene Analyse und Entwurf)		
Testfalltabelle	Akzeptanztest	Unit-Test	
Java + Standardbibliothek	-	-	x
junit (3.8.1)	-	-	x

**Beachte:**

- dass die einzelnen Artefakte verschiedene Sichten auf ein- und dasselbe Softwaresystem darstellen!
- Deshalb müssen die Zusammenhänge zwischen den Artefakten verstanden und auf deren Konsistenz geachtet werden!
- Der Übergang von aUML zu (dUML und) jUML bedeutet im Allg. Verfeinerung oder anders gesagt den Übergang vom Problemraum zum Lösungsraum.

<sup>1</sup> In den Übungsaufgaben unterscheiden wir aus Komplexitätsgründen nicht zwischen dUML und jUML. Wir entwerfen gleich in jUML. D.h. wir gehen davon aus, dass unser Entwurf für Java-Programme gemacht ist: z.B. Angabe von Java-Datentypen, nur Einfachvererbung, Arbeit mit Interfaces usw.

<sup>2</sup> Ein einzelner Anwendungsfall sollte durch eine Tabelle oder weitere UML-Diagramme (meistens Aktivitätsdiagramm oder Sequenzdiagramm) beschrieben/verfeinert werden.

<sup>3</sup> „-“ bedeutet: wird nicht angewendet

<sup>4</sup> Objektlebenszyklus (OLC), Anwendungsfalllebenszyklus (ULC), Steuerungsmaschine (CSM)

