

# M 226 a Klassenbasiert (ohne Vererbung) implementieren

## Kompetenzraster M226ab

Hz 1	Ein SW-Design mit Klassen nachvollziehen und mit eigenen fachlichen und technischen Klassen ergänzen.	<b>A1</b> gesch. Aufwand L
	<ol style="list-style-type: none"><li>1 Zeigen Sie die Abstraktionsmittel wie Beziehungen, Klassen, Attribute und Methoden, mit denen die reale Welt im Design abgebildet wird.</li><li>2 Zeigen Sie, wie die Klasse als abstrakten Datentyp (ADT) zu verstehen ist.</li><li>3 Zeigen Sie die Beziehungstypen wie Assoziation, Aggregation, Komposition zwischen Klassen.</li><li>4 Zeigen Sie die Bedeutung von Fachklassen und Klassen, die für die Implementation der Anwendung nötig sind und das klassenbasierte Design vervollständigen (technische Klassen).</li><li>5 Zeigen Sie das Zusammenspiel mehrerer Klassen in einer Anwendung und wenden Sie das Prinzip der Delegation an.</li></ol>	
Hz 2	Die Notation dynamischer und statischer Strukturen einer Anwendung mittels Unified Modeling Language (UML) nachvollziehen.	<b>A2</b> gesch. Aufwand S
	<ol style="list-style-type: none"><li>1 Zeigen Sie Diagrammnotationen zur Darstellung von Klassen und Objekten.</li><li>2 Zeigen Sie Diagrammnotationen zur Darstellung des Meldungs-austausches zwischen Objekten.</li></ol>	
Hz 3	Klassenbasiertes Design implementieren.	<b>A3</b> gesch. Aufwand M
	<ol style="list-style-type: none"><li>1 Zeigen Sie eine objektorientierte Sprache, welche für die Umsetzung des klassenbasierten Designs geeignet ist.</li><li>2 Zeigen Sie das Prinzip und die Anwendung des Information Hiding und der Kapselung beim Entwurf von Klassen..</li><li>3 Zeigen Sie den Unterschied zwischen dem Programm zur Kompilationszeit (Klassen) und zur Laufzeit (sich referenzierende Objekte).</li></ol>	
Hz 4	Für funktionale Einheiten einer Anwendung Testfälle implementieren, um die Anwendung automatisch zu prüfen.	<b>A4</b> gesch. Aufwand M
	<ol style="list-style-type: none"><li>1 Zeigen Sie Vorgehen um für Einheiten Testfälle mittels geeigneter Grenzwerte oder Pfadprüfungen zu finden und festzulegen.</li><li>2 Zeigen Sie die zur Verfügung gestellten Mittel einer Testinfrastruktur um die Testfälle zu implementieren.</li><li>3 Zeigen Sie Mittel, die Abdeckung der Tests zu messen.</li></ol>	
Hz 5	Klassen der Anwendung systematisch, unter Verwendung der hierfür vorgesehenen Infrastruktur, dokumentieren.	<b>A5</b> gesch. Aufwand S
	<ol style="list-style-type: none"><li>1 Zeigen Sie die Möglichkeiten der Dokumentationsinfrastruktur und zeigen Sie wie die Programmeinheiten gemäss Vorgaben zu dokumentieren sind.</li><li>2 Zeigen Sie den Nutzen von Api-Dokumentationen und zeigen Sie wie diese erzeugt werden.</li></ol>	

# M 226 b objektorientiert (mit Vererbung) implementieren

## Kompetenzraster M226ab

Hz 1	Ein objektorientiertes Design nachvollziehen und durch Einsatz der Vererbung erweitern.	<b>B1</b> gesch. Aufwand <b>L</b>
	<ol style="list-style-type: none"><li>1 Zeigen Sie den objektorientierten Ansatz mit Klassen und Vererbung.</li><li>2 Zeigen Sie wie Sie durch die Anwendung von Vererbung Redundanzen im Klassendesign eliminieren.</li><li>3 Zeigen Sie wie Sie durch den Einsatz von Delegation und Vererbung switchartige Strukturen vermeiden und dadurch die Erweiterbarkeit der Anwendung sicherstellen.</li><li>4 Zeigen Sie, wie man das Prinzip der dynamischen Bindung verstehen kann.</li><li>5 Zeigen Sie, wie man beim Erstellen von Vererbungshierarchien das Prinzip von Spezialisierung/Generalisierung inklusive abstrakter Klassen. berücksichtigt</li></ol>	
Hz 2	Die Notation dynamischer und statischer Strukturen einer Anwendung mittels Unified Modeling Language (UML) nachvollziehen.	<b>B2</b> gesch. Aufwand <b>S</b>
	<ol style="list-style-type: none"><li>1 Zeigen Sie Diagrammnotationen zur Darstellung von Vererbung.</li><li>2 Zeigen Sie Diagrammnotationen zur Darstellung von Schnittstellen.</li></ol>	
Hz 3	Objektorientiertes Design implementieren.	<b>B3</b> gesch. Aufwand <b>M</b>
	<ol style="list-style-type: none"><li>1 Zeigen Sie Elemente einer objektorientierten Sprache für die Umsetzung der Vererbung im objektorientierten Design.</li><li>2 Zeigen Sie Elemente einer objektorientierten Sprache für das Überschreiben von Methoden im objektorientierten Design.</li><li>3 Zeigen Sie Elemente einer objektorientierten Sprache für die Realisierung von Schnittstellen und abstrakten Klassen.</li></ol>	
Hz 4	Fortgeschrittene Testfälle für funktionale Einheiten implementieren, welche durch geeignete Techniken von anderen Systemteilen unabhängig sind.	<b>B4</b> gesch. Aufwand <b>M</b>
	<ol style="list-style-type: none"><li>1 Zeigen Sie das Hilfsmittel des Mockings um Abhängigkeiten aus den Testfällen zu eliminieren.</li><li>2 Zeigen Sie Mittel zur Isolierung der Testdaten und Testfälle, um die gegenseitige Unabhängigkeit der Testfälle zu garantieren.</li><li>3 Zeigen Sie Werkzeuge zur Implementation von isolierten Testfällen.</li></ol>	