

# ContactApp mit Java - Bewertung zum Modul 320 - Objektorientiert Programmieren - 100% der Modulnote

## Situation:

Die Lernenden implementieren die Applikation ContactApp *als Einzelarbeit* in einem modulübergreifenden Projekt, welches einer agilen Projektmethodik folgt. Die Anforderungen an die ContactApp werden in Form von Requirements, User Stories und Use Cases beschrieben. Diese Arbeiten, welche im Rahmen der Analyse stattfinden, erledigen die Lernenden gemeinsam. Auch die Sprint-Planung erfolgt im Plenum. Während der Implementierung arbeiten die Lernenden weitgehend selbständig und sorgen dafür, dass die Sprint-Ziele erreicht werden. Dadurch bestimmen sie den Projektverlauf nachhaltig und ermöglichen quantitativ und qualitativ gute Ergebnisse. Hinsichtlich des Quellcodes wird eine Eigenleistung von mindestens 66% verlangt, was mit JPlag ausgewertet wird. *Unzureichende Eigenleistung, verspätete Abgabe, nicht nachvollziehbare Ergebnisse, unerlaubter Einsatz von künstlicher Intelligenz oder Plagiarismus resultieren ausnahmslos in der Note 1.*

Kandidat		Selbsteinschätzung				Total: 90 Punkte; Note 6 mit 82 Punkten.	Bewertung durch die Lehrperson			
Nachname, Vorname		trifft überhaupt nicht zu	trifft eher nicht zu	trifft zu	trifft in hohem Masse zu	1.0	trifft überhaupt nicht zu	trifft eher nicht zu	trifft zu	trifft in hohem Masse zu
Nr.	Kriterium	--	-	+	++	Bemerkungen, Anregungen aus der Besprechung	-- (0P)	- (1P)	+ (2P)	++ (3P)
1	Ich kann die Projektstruktur nach Vorgabe (dem Projekt schneller) einrichten.									
2	Ich kann die Unterrichtszeit gewinnbringend und effizient einsetzen.									
3	Ich kann gitlab und git professionell anwenden. Ich führe ein .gitignore. Ich kann mit Branches arbeiten und weise wöchentlich mindestens einen commit aus. Anhand der Commit-Message kann nachvollzogen werden, was gemacht wurde.									
4	Ich kann Naming Conventions konsequent durchsetzen.									
5	Ich kann in Java eigene Klassen mit Eigenschaften und Verhalten schreiben.									
6	Ich kann Variablen mit einfachen und abstrakten Datentypen situativ richtig anwenden.									
7	Ich kann Kontrollstrukturen (if, else, for, while ...) situativ richtig anwenden.									
8	Ich kann ArrayLists und Generics situativ richtig anwenden.									
9	Ich kann Methoden mit Parametern und/oder Rückgabewerten situativ richtig anwenden.									
10	Ich kann Events verarbeiten (z.B. ActionEvent, MouseEvent).									
11	Ich kann bei meinen eigenen Klassen das Single Responsibility Principle (SRP) anwenden.									
12	Ich kann bei meinen eigenen Klassen das Liskov Substituion Principle beachten. Damit kann ich auch aufzeigen, wann eine Vererbung angezeigt ist und wann nicht.									
13	Ich kann bei meinen eigenen Klassen Dependency Inversion (Inversion of Control) anwenden. Damit vermeide ich unnötige Abhängigkeiten unter meinen Klassen.									
14	Ich kann die Applikation nach der MVC-Architektur gestalten.									
15	Ich kann in meiner Applikation das Observer Pattern mit eigenen Klassen und Interfaces gewinnbringend anwenden. Aus dem Javadoc kann entnommen werden, wo ich das Observer Pattern angewendet habe.									
16	Ich kann in meiner Applikation das Strategy Pattern gewinnbringend anwenden. Aus dem Javadoc kann entnommen werden, wo ich das Strategy Pattern angewendet habe.									
17	<b>Zählt sechsfach (25% der Maximalpunktzahl): Ich kann die Sprint-Ziele erreichen oder übertreffe die gesetzten Ziele.</b>									
18	Ich kann einen Testfall (Blackbox) nach Vorgabe korrekt beschreiben.									
19	Ich kann einen Testfall (Blackbox) nach Vorgabe korrekt durchführen und protokollieren.									
20	Ich kann einen Unit-Test schreiben.									
21	Ich kann einen Unit-Test ausführen.									
22	Ich kann mindestens drei wesentliche Klassen mit Javadoc dokumentieren. Die generierte Dokumentation lege ich unter <i>doc/javadoc</i> ab.									
23	Ich kann für meine fachlichen Klassen ein mit drawio eigenhändig erstelltes Klassendiagramm darlegen.									
24	Ich kann sicherstellen, dass die Lehrperson mit meiner Lieferung alles bekommt, was sie zur Beurteilung des Projekts braucht. Der Quellcode und alle Artefakte sind im main-branch enthalten.									
25	Ich kann diese Selbsteinschätzung reflektiert vornehmen. Ich weiss also, was ich gut kann und was noch nicht so gut klappt.									
<b>Formatives Feedback</b>							0Punkte			