

Übung 05

Korrektur: Sebastian Pritz

Name: Fabian Tober

Punkte: 53

Allgemeine Anmerkungen:

Beispiel 1: Flugreisen

24 von 30

Lösungsidee: 3/3

Anmerkung	Abzug
Okay.	

Quellcode: 11/17

Anmerkung	Abzug
Es sollte für jede Klasse nicht nur ein eigenes .h File geben, sondern auch ein eigenes .cpp. Anders, also mit allen Klassen in 2 Files (1x h, 1x cpp), macht es nur in sehr speziellen Situationen, wie z.B. der Komposition (in Teil zwei, mit den Parts) Sinn. Das hier ist etwas merkwürdig. Die Funktion new_person ist eigentlich ein überladener Konstruktor, und so wie jetzt implementiert, eigentlich fehl am Platz.	-2
Kein einziges Kommentar, bis auf die neben den Konstruktoren.	-2
Sonst aber okay 😊	-2

Testfälle: 10/10

Anmerkung	Abzug
Passt!	

Beispiel 2: Stücklistenverwaltung

29 von 70

Lösungsidee: 4/7

Anmerkung	Abzug
Die Lösungsidee ist etwas oberflächlich bzw. wiederholt nur das, was schon gegeben ist. → Auf was ist aufzupassen? Wie würdest du die Implementierung der Formatter bzw. der Store und Load Methoden implementieren? Was wird bei „equals“ überprüft? Es geht wirklich darum, über eine Lösung nachzudenken, und nicht nur das Problem bzw. Aufgabenstellung zu wiederholen.	-3

Quellcode: 16/40

Anmerkung	Abzug
Ist mir vorher schon aufgefallen, aber: macht es Sinn, Parts (bzw. vorher Personen) mit dem Default Konstruktor zu erstellen? Es wird praktisch keine Information gehalten und ist bis zur Initialisierung (die ja gerade im Konstruktor geschehen sollte!) nur Speicherverschwendung. Instanzen von Klassen sollte man normalerweise mittels dem „new“ Schlüsselwort erstellen (Part* a = new Part(„Sessel“)), sonst existiert die Instanz nur in der Methode und das kann zu merkwürdigem Verhalten führen. Formatter probiert aber funktionieren nicht und Code ergibt auch relativ wenig Sinn bzw. erfüllt nicht die gezeigten Anforderungen. Warum wird z.B. jeder Sub-Part mit dem eigentlichen Part mit „equals“ verglichen?... Store und Load nicht implementiert Widerum keine Kommentare. Hätte unter Umständen geholfen das Problem zu isolieren bzw. Übersicht zu schaffen.	Hinweis -2 -12 -10 -2

Testfälle: 11/23

Anmerkung	Abzug
Formatter nicht getestet, bzw. nichts in Doku vermerkt	-4
Lesen/Schreiben nicht getestet	-6
Kein komplexes Beispiel wie z.B. in der Angabe getestet (mind. 2 Layers von CompositePart)	-2