

Übung 05

Korrektur: Sebastian Pritz

Name: Sigrid Klein

Punkte: 89

Allgemeine Anmerkungen:

Beispiel 1: Flugreisen

28 von 30

Lösungsidee: 3/3

Anmerkung	Abzug
Auch wenn die Klassen relativ klein sind, geht man normalerweise den Weg und teilt diese in .cpp und .h auf. In diesem Fall aber okay, weil explizit erwähnt. Sonst alles gut beschrieben 😊	Hinweis

Quellcode: 15/17

Anmerkung	Abzug
Alle Klassen in einem einzigen File zu implementieren ist normalerweise bad practice. Hier natürlich kein großes Problem, weil das Projekt mini ist, aber bei großen Projekten ist es sehr hilfreich, wenn Klassenname == Filename.	-2

Testfälle: 10/10

Anmerkung	Abzug
Kurz, klar und alles dabei. Passt! 😊	

Beispiel 2: Stücklistenverwaltung

61 von 70

Lösungsidee: 7/7

Anmerkung	Abzug
„Store speichert in Vektor, Load schreibt Vektor auf Konsole“ → Parts/CompositeParts sollen <i>persistent</i> , d.h. über die Laufzeit des Programms hinaus, gehalten werden können. D.h. die Definition oben würde das nicht erfüllen.	Hinweis

Quellcode: 33/40

Anmerkung	Abzug
Ein Interface stellt nur Methoden bereit (= Schnittstelle, welche die Klasse erfüllen muss), keine zusätzlichen Attribute etc. → Storable ist hier kein Interface, sondern eine normale Klasse von der abgeleitet wird.	-2
Die Funktion von store/load ist, wie in der Korrektur oben erwähnt, so nicht korrekt. Das entspricht eher einer Konverter-Methode wie z.B. „part.to_vector()“ oder sowas, und warum „load“ den umgewandelten Vektor ausgibt - da verstehe ich den Zusammenhang mit dem Namen bzw. der Angabe nicht? → CompositeParts in File schreiben, und mithilfe eines Files erstellen wäre gefordert gewesen. (in File <i>schreiben</i> und aus File <i>laden</i>)	-5

Testfälle: 21/23

Anmerkung	Abzug
Equals nicht getestet	-2