

## Zusammenführung Heldenprojekt

Nach der doch recht langen aufgeteilten Arbeit an dem Heldenprojekt beginnt nun das Zusammenfügen der vorhandenen Teile. Wenn die Schnittstellenvereinbarungen gut waren und wirklich 1:1 umgesetzt worden sind / wären, so würde jetzt kaum Arbeit auf uns zukommen. Da das aber nicht ganz so genau eingehalten wurde (vor allem Punkt zwei), ist hier noch etwas Arbeit zu leisten. Diese Arbeit wird von je maximal zwei Personen vorgenommen (also ist die Arbeit in den großen Gruppen vorbei).

Damit dennoch Hilfestellungen vorliegen, geben wir Ihnen hiermit ein paar Tipps, welche sie verwenden können, um die Klassen zusammenzufügen.

### 0) Vorarbeit

- a. Ein neues Projekt mit BlueJ anlegen (merken wo!).
- b. BlueJ schließen.
- c. Die Dateien unter Schuelerdaten/Adventuregame in den Ordner des neu erstellten Projektes kopieren.
- d. BlueJ starten und falls noch nicht geschehen das Projekt wieder öffnen.

### 1) Erstes Übersetzen, Fehler die dabei auftreten:

- a. `public void waffeAufnehmen (Waffe w) ;` → was macht denn hier das Semikolon?
- b. `rucksack.append (w);` → schauen sie sich doch mal die Waffenliste an, welche Methoden zum Anfügen gibt es da? (`append` kann nicht verwendet werden) Diese Fehlermeldung gibt es häufiger.
- c. Im Schmied: `eintrag = auftragsBuch.dequeue();` gibt einen Fehler, dass hier eigentlich ein Eintrag erwartet wird. Vielleicht ist hier `dequeue` nicht so ganz richtig gewählt?
- d. Im Schmied: `if (w.gibZustand() < 100 && w.gibZustand() > 0)` `w` ist nicht bekannt. `w` ist ja in dem `eintrag` gespeichert. Also muss man auch über ihn an `w` heran. Dazu muss der Eintrag um `get` Methoden für die privaten Attribute erweitert werden: `Waffe getWaffe()`, `Held getHeld()`, und `int getAnteil()`. Dann kann man mit `eintrag.getWaffe().gibZustand()` auf den Zustand der Waffe zurück greifen. (selbes mit `Prozent` → wo steht das?) verändere / veraendere sollte einheitlich geschrieben werden.
- e. `eintrag = new Eintrag(prozent, Held, Waffe);` → was sind hier Held und Waffe? Wie heißen die übergebenen Parameter wirklich?

### 2) Jetzt sollte das Übersetzen keine Probleme mehr bereiten. Aber:

- a. die Waffe kann noch keinen Namen haben, dabei erbt sie den eigentlich von Gegenstand → was kann man da machen?
- b. Will man dem Schmied etwas in auftrag geben, so erhält man eine `null Pointer Exception`: `auftragsBuch.enqueue(eintrag);` → gibt es das `AuftragBuch` zu diesem Zeitpunkt bereits?
- c. Da die Methode `reparieren` privat ist, kann sie nicht von außen ausgeführt werden. Der Schmied repariert immer dann, wenn er bezahlt wird. Erweitern sie das Modell um die Möglichkeit des Bezahlens.