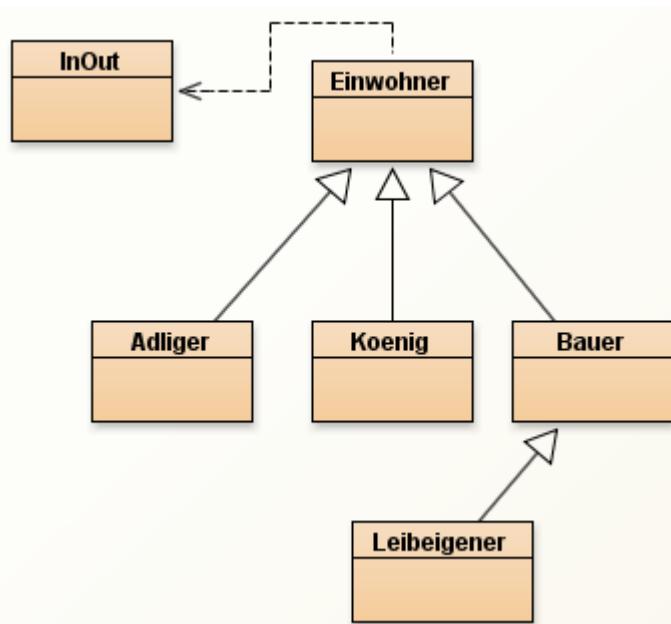


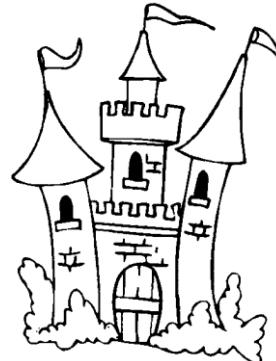


Übung Vererbung und Polymorphie

In einem mittelalterlichen Königreich soll das Finanz- und Steuerwesen auf Computer-Technik umgestellt werden. Die verschiedenen Bevölkerungsgruppen werden durch die folgende Klassenhierarchie modelliert:



Einwohner
einkommen: int # steuer: int # name: String # stand: String
+ Einwohner() # steuerBerechnen(): void # steuerAusgabe(): void



Das Attribut `einkommen` gibt das tatsächliche Jahreseinkommen des Einwohners in Tälern an.

Die Methoden `steuerBerechnen()` und `steuerAusgabe()` sollen für jeden Einwohner des Königreiches korrekte Werte gemäß der königlichen Vorschriften liefern:

1. Sofern dieses Gesetz nichts Anderes aussagt, hat jeder Einwohner sein gesamtes Einkommen zu versteuern.
2. Jeder Einwohner hat 10% seines zu versteuernden Einkommens als Steuer zu entrichten. Der Steuerbetrag wird auf ganze Taler gerundet, jedoch beträgt die Steuer immer mindestens 1 Taler.
3. Der König zahlt *keine* Steuern.
4. Für Angehörige des Adels beträgt die Steuer mindestens 20 Taler.
5. Bei Leibeigenen sind 12 Taler des Einkommens steuerfrei.

Da jährliche Änderungen bei der Steuerberechnung zu erwarten sind, darf die Grundregel (2.) - änderungsfreundlich- nur an einer Stelle der Klassenhierarchie implementiert werden.

Weiterhin gilt:

Bei der Ausgabe sollen Namen, Stand und Einkommen und Steuer ausgegeben werden. Der Stand wird für jedes Objekt einer Klasse automatisch im Konstruktor instanziert.

Programmieren Sie die Klassenhierarchie! Implementieren Sie zuerst die Basisklasse. Welche Methoden müssen in den Subklassen überschrieben werden?

