U02 Einführung in Java (II)

Inhalt der Übung

- Schreiben von Methoden (Beispiel `HelloLibrary` aus Übung U01)
- Vererbung und Polymorphie (Beispiel `Koenigreich`)

Übungsaufgaben

Aufgabe 1

Erweitern Sie das `HelloLibrary`-Programm aus Übung U01 um die Ausleihe eines Buches. Gehen Sie dabei wie folgt vor:

- Schreiben Sie für die Klasse `Library` eine Methode `search()`, die prüft, ob das gesuchte Buch (mit einem bestimmten Titel) in der Bibliothek registriert ist.
- Implementieren Sie für die Klasse `Book` einen Ausleihstatus (`state`).
- Schreiben Sie für die Klasse `Library` eine Methode `loan()`, die ein Buch mit einem bestimmten Titel ausleiht. Ein Benutzer soll nicht berücksichtigt werden!

Überlegen Sie sich jeweils

- um welche Attribute und Methoden Sie die Klassen erweitern müssen
- wie Ergebnistyp und Signatur (Methodenname, Parameterliste) der jeweiligen Methode aussehen
- die Implementation des Methodenrumpfes

Testen Sie das Programm in Ihrer eigenen Java-Entwicklungsumgebung!

Aufgabe 2

In einem mittelalterlichen Königreich soll das Finanz- und Steuerwesen auf EDV umgestellt werden. Die verschiedenen Bevölkerungsgruppen werden durch die Klassenhierarchie auf Seite 2 modelliert.

Das Attribut `einkommen` gibt das tatsächliche Jahreseinkommen des Einwohners in Talern an. Die Methoden `zuVersteuerndesEinkommen()` und `steuer()` sollen die für jeden Einwohner des Königreiches korrekte Werte gemäß der folgenden königlichen Vorschriften liefern:

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

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

- Implementieren Sie die Klassenhierarchie! Überlegen Sie sich zunächst, wie die Methoden in der Klasse `Einwohner` implementiert werden müssen. Welche Methoden müssen in den Unterklassen überschrieben werden? Wie ergänzt man das obige Klassendiagramm?
- Überlegen Sie sich, welche Ausgabe `main()` der Klasse `Koenigreich` (`Koenigreich.java`) erzeugt!
- Testen Sie zu Hause das Programm! In INLOPP ist das die Aufgabe `Inheritance (Middle Age Kingdom)`.