

Modellierung und Programmierung 1 Übungsserie 4

Lösungsvorschläge

1. Rekursion

- a) Die Fakultät von natürlichen Zahlen wird bereits für kleine Zahlen so groß, dass sie mit **long** nicht mehr darstellbar ist. Deshalb multipliziert und dividiert man im Wechsel. Um Fehler bei der ganzzahligen Division zu vermeiden, muss man bei der iterativen Lösung die Iteration mit dem größten Zähler und dem kleinsten Nenner beginnen.

Siehe Methode **binom1** im Programm **Binom.java**

- b) Siehe Methode **binom2** im Programm **Binom.java**

- c) Siehe Methode **binom3** im Programm **Binom.java**

- d) Ergebnis für $\binom{19}{17}$ und für $\binom{19}{13}$:

Methode	binom1	binom2	binom3
Ergebnis 19 über 17	171	171	171
Anzahl Aufrufe	1	341	3
Ergebnis 19 über 13	27132	27132	27132
Anzahl Aufrufe	1	54263	7

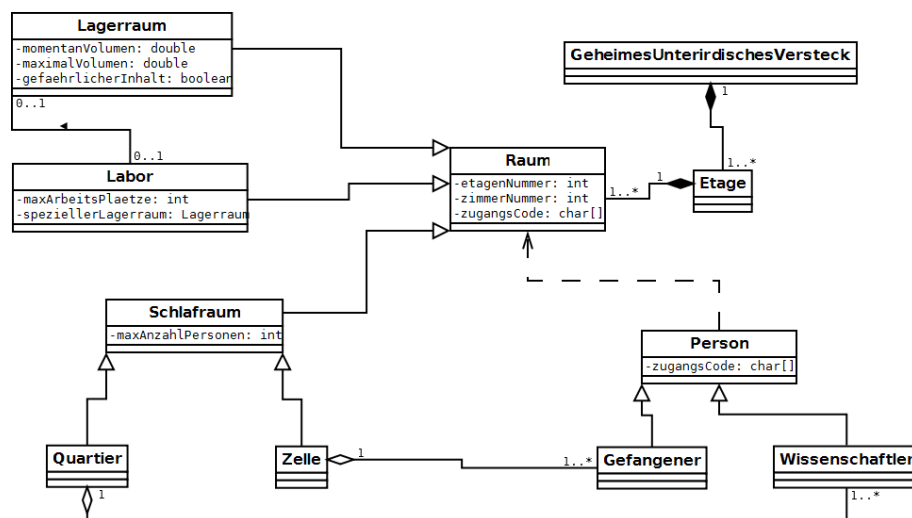
Die Methoden **binom1** und **binom3** sind beide erheblich effizienter als **binom2**. Insbesondere berechnet **binom2** Ergebnisse mehrfach. Beispielsweise wird für $\binom{7}{3}$ und $\binom{7}{2}$ jeweils unabhängig voneinander $\binom{6}{2}$ berechnet, was wiederum andere Koeffizienten mehrfach berechnet und so weiter. Der Rechenaufwand für die Methoden **binom1** bzw. **binom3** ist grob durch k (für $k < (n - k)$) beschränkt.

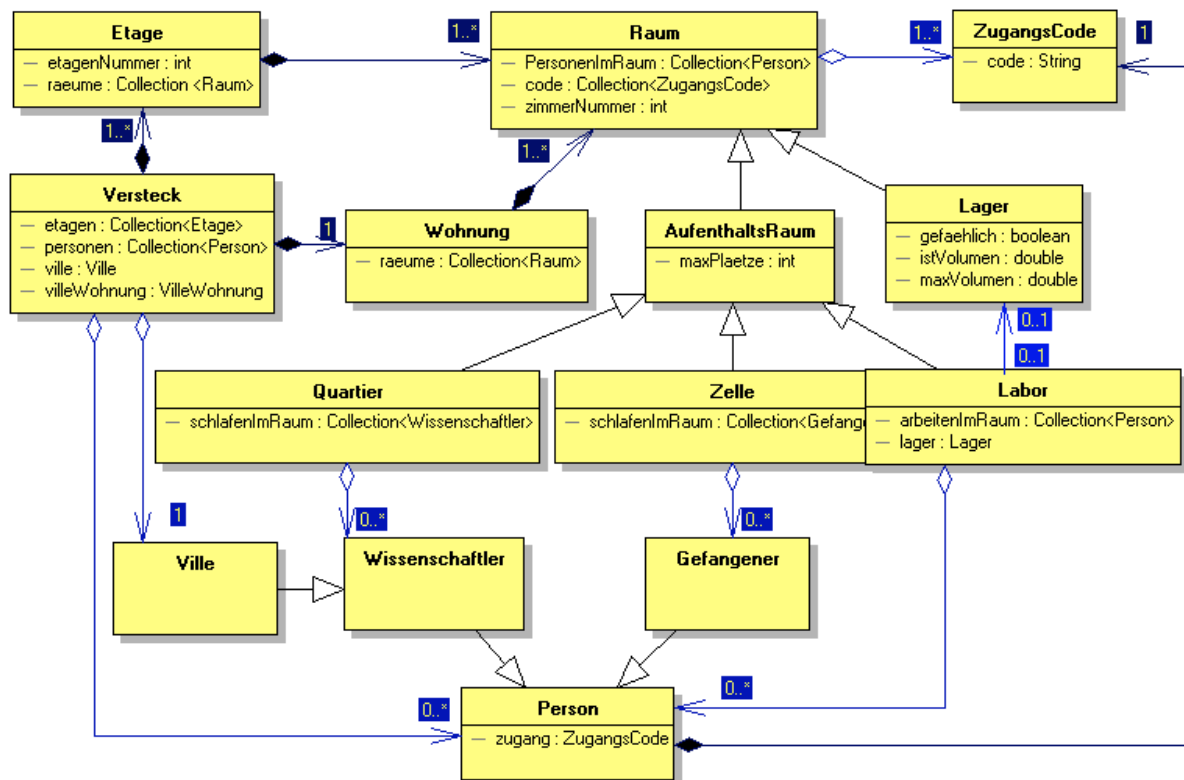
2. Klassen

- a) bis d) siehe Klasse **MitSpieler.java** und Online-Dokumentation in **MitSpielerDoc**.

3. Modellierung

- a) Zwei Varianten:





b) Zur zweiten Variante:

Beziehung	Begründung
Komposition Versteck → Etage	Ein Versteck hat mindestens eine Etage (1..*), eine Etage existiert nur in einem Gebäude (1).
Aggregation Versteck → Person	Im Versteck befinden sich evtl. Personen (0..*), eine Person ist im Gebäude oder nicht (0..1).
Komposition Versteck → Wohnung	Das Versteck hat eine Wohnung (1), eine Wohnung existiert nur in einem Gebäude (1).
Aggregation Versteck → Ville	Im Gebäude wohnt Ville (1), Ville befindet sich im Gebäude oder nicht (0..1).
Komposition Etage → Raum	Eine Etage hat mindestens einen Raum (1..*), ein Raum existiert nur in einer Etage (1).
Komposition Wohnung → Raum	Eine Wohnung hat mindestens einen Raum (1..*), ein Raum ist von einer Wohnung existenzabhängig (1).
Aggregation Quartier → Wissenschaftler	In einem Quartier schlafen Wissenschaftler (0..*), ein Wissenschaftler schläft gerade oder nicht (0..1).
Aggregation Zelle → Gefangener	In einer Zelle schlafen Gefangene (0..*), ein Gefangener schläft gerade oder nicht (0..1).
Aggregation Labor → Person	In einem Labor arbeiten Personen (0..*), eine Person arbeitet gerade oder nicht (0..1).
Komposition Person → ZugangsCode	Eine Person hat genau einen Zugangscode (1), ein Zugangscode existiert nur in Verbindung mit einer Person (1).
Aggregation Raum → ZugangsCode	Ein Raum speichert den Zugangscode von Personen (0..*), ein Zugangscode ist dem Raum bekannt oder nicht (0..1).
Assoziation Labor → Lager	Ein Labor hat manchmal ein Lager (0..1), ein Lager kann zu einem Labor gehören (0..1).

Hinweis:

An der Raute kann bei Aggregation (0..1) und bei Komposition (1) weggelassen werden.