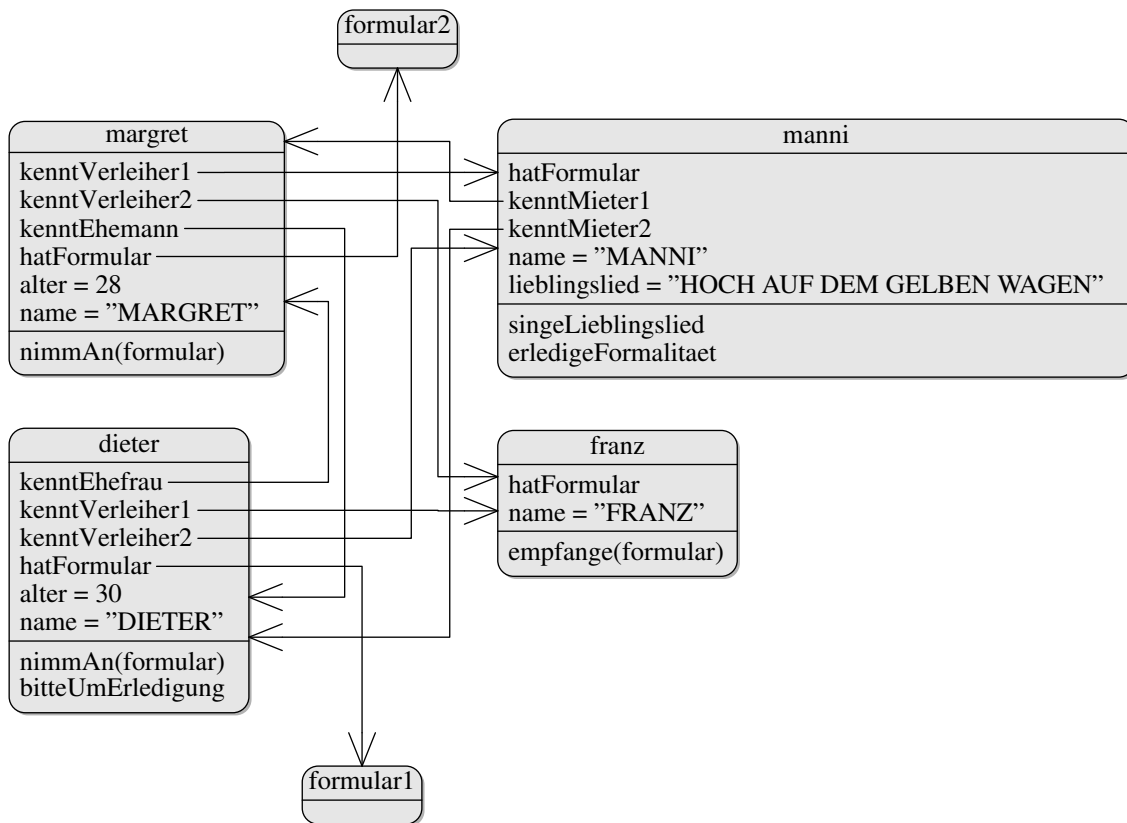


# 1 Modellierung mit Objekten



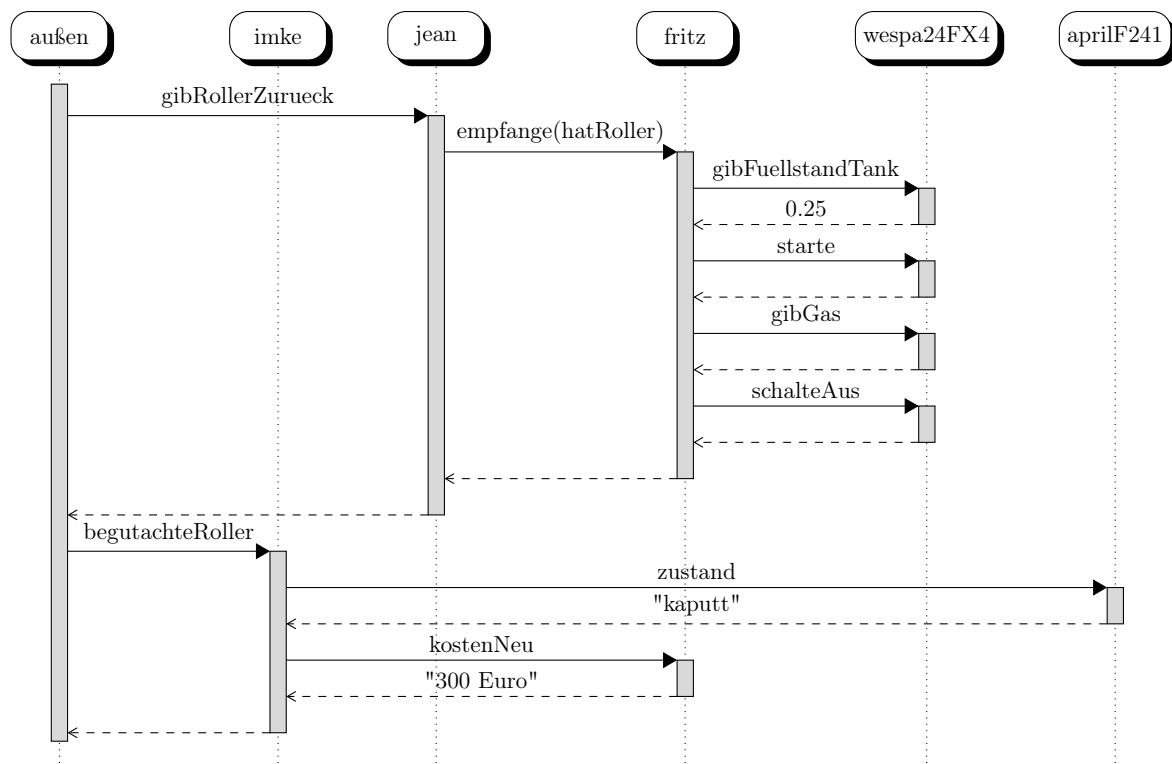
Die Punkteverteilung:

- 6 Objekte → 3 Punkte.
- Attribute: `alter`, `hatFormular`, `lieblingslied` je 1P, die `kennt`-Beziehungen zusammen 3 Punkte.
- Methoden: `3 · nimmAn(formular)` gibt 3 Punkte (variable Verteilung), `erledigeFormalitaet` und `singeLied` jeweils einen.
- Konventionseinhaltung: Objektnamen klein, Attributwerte groß und Aufträge im Imperativ jeweils 1P. Präfixe bei den Attributen → 2P.



## 2 Sequenz- und Klassendiagramme

### Teil a

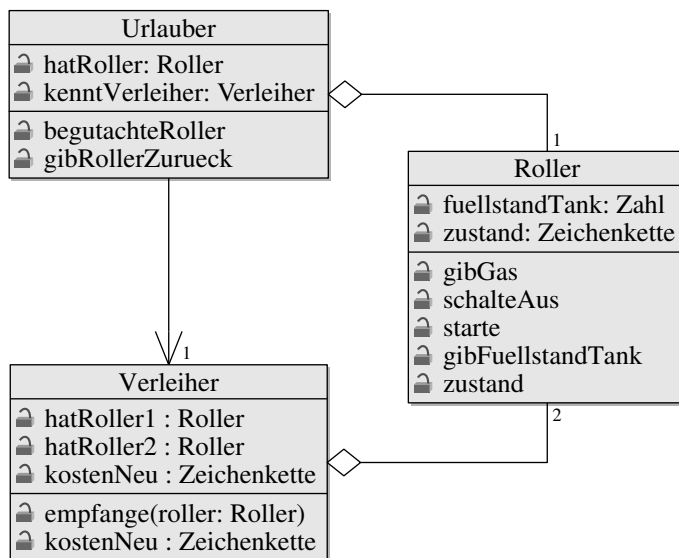


Die Punkteverteilung:

- Für jede Nachricht/Antwort einen Punkt. Auf " achten!



## Teil b



Die Punkteverteilung:

- 3 Klassen → 1P
- Attribute: 6 Punkte +1, wenn die Kardinalität 2 bei Verleiher → Roller erkennbar ist
- Methoden: 7 Punkte (`starte`, `gibGas`, `schalteAus` zusammen nur einen Punkt), Namenskonventionen: 2 Punkte

## Teil c

```

1 def empfangen (self , roller):
2     self.hatRoller= roller
3     print (roller.gibFuellstandTank())
4     roller.starte()
5     roller.gibGas()
6     roller.schalteAus()
  
```

Die Punkteverteilung:

- `starte`, `gibGas`, `schalteAus` zusammen einen Punkt.
- Handling der Anfrage einen Punkt.
- Klassennamen jeweils einen Punkt
- Ein Punkt für die Punktschreibweise.
- Wer `self.hatRoller= roller` dabei geschrieben hat, konnte sich noch einen Extra-Punkt verdienen.

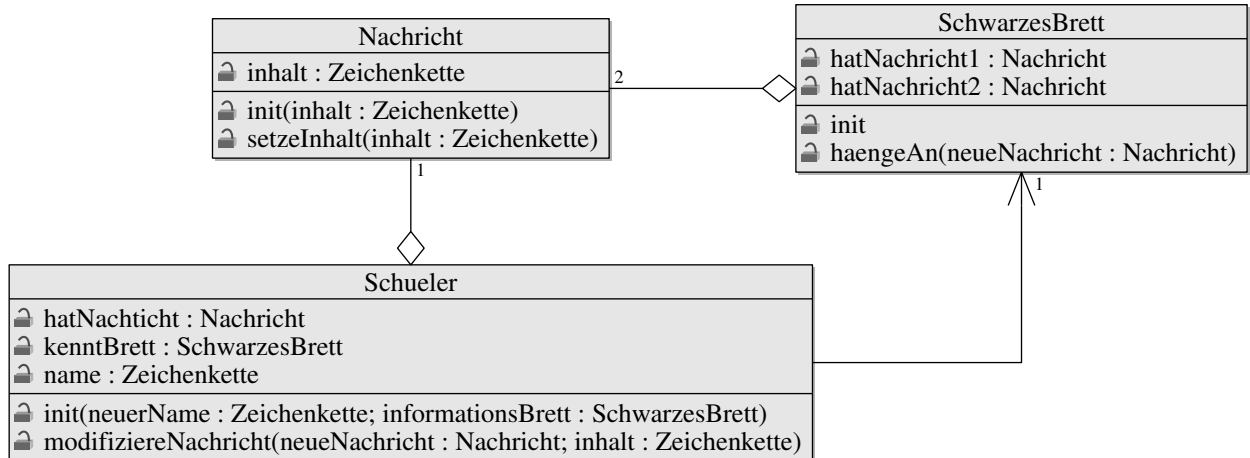


### 3 Arbeiten mit Quellcode

#### Teil a

- Objekterzeugung und Methode müssen drin sein. 2 Punkte

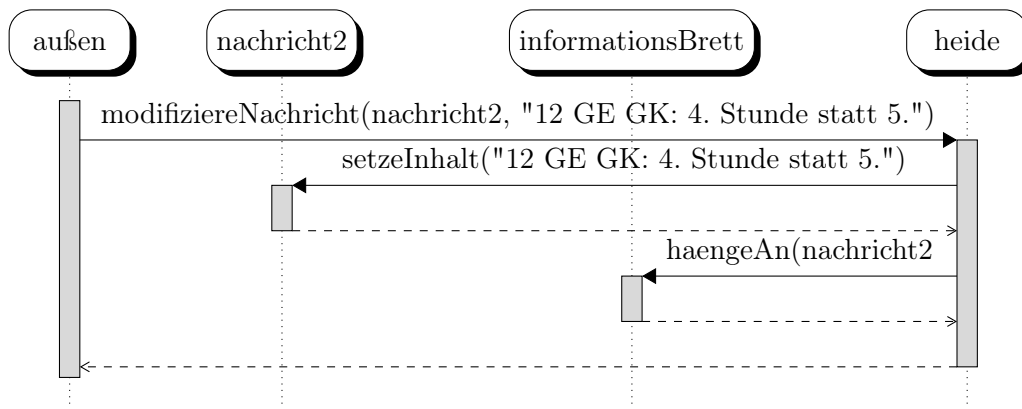
#### Teil b



Punkteverteilung:

- 3 Klassen → 1 Punkt.
- Attribute: 6 Punkte
- Methoden: 6 Punkte + 3 für die Parameter

#### Teil c



Punkteverteilung:

- 3 Objekte → 1



- 3 Nachrichten →3
- Zuweisungen ignoriert →1
- Parameter angegeben →1

