

## Aufgabe 1 (12 Punkte)

### a. Klasse Bus (7 Punkte)

Erweitern Sie Ihr Fuhrpark-Programm um eine neue konkrete Klasse *Bus*, die von der abstrakten Klasse *Fahrzeug* abgeleitet wird. Die Klasse *Bus* soll die gleichen Instanzvariablen wie die Klasse *Pkw* besitzen.

Der Rabatt für Busse wird (ähnlich zu PKWs) abhängig vom Fahrzeugalter und Servicejahr berechnet: pro Jahr einen Grundrabatt von **3%** plus **3%** pro Jahr seit dem letzten Service; der maximal mögliche Rabatt beträgt für Busse **25%**; das Minimum 0%.

Erweitern Sie Ihr Programm damit alle Funktionen auch für Busse zur Verfügung stehen.

- Parameter 'add **bus** <id> <marke> <modell> <baujahr> <grundpreis> <ueberpruefungsdatum>'
  - BUS persistent hinzufügen

Beispielaufruf:

```
java FahrzeugClient <Datei> add bus 7 Mercedes Citaro 2010 50000 2014
```

Beispielaufruf:

```
java FahrzeugClient <Datei> show
```

Ausgabe:

```
Typ:      PKW
Id:       5
Marke:    Tesla
Modell:    Model S
Baujahr:   2016
Grundpreis: 65000.00
Servicejahr: 2016
Preis:     55250.00
```

```
Typ:      PKW
Id:       2
Marke:    Opel
Modell:    Manta
Baujahr:   1972
Grundpreis: 21000.00
Servicejahr: 2013
Preis:     17850.00
```

```
Typ:      LKW
Id:       3
Marke:    MAN
Modell:    TGX 6X2
Baujahr:   2014
Grundpreis: 56763.00
Preis:     45410.40
```

```
Typ:      LKW
Id:       22
Marke:    Steyr
Modell:   990
Baujahr:  1972
Grundpreis: 6900.00
Preis:    5520.00
```

```
Typ:      BUS
Id:       7
Marke:    MAN
Modell:   Citaro
Baujahr:  2010
Grundpreis: 50000.00
Servicejahr: 2014
Preis:    37500.00
```

```
Typ:      BUS
Id:       7
Marke:    MAN
Modell:   City
Baujahr:  2018
Grundpreis: 100000.00
Servicejahr: 2019
Preis:    91000.00
```

- Parameter 'count **bus**'
  - Gesamtzahl der Busse berechnen

Beispielaufruf:

```
java FahrzeugClient <Datei> count bus
```

Ausgabe:

```
2
```

Beispielaufruf:

```
java FahrzeugClient <Datei> meanprice
```

Ausgabe:

```
42088.40
```

## **b. Alle Fahrzeuge innerhalb eines Preisbereichs ausgeben (5 Punkte)**

Erweitern Sie Ihr Programm damit alle Fahrzeuge, deren Preis sich innerhalb des angegebenen Bereichs befindet, ausgegeben werden.

- Parameter '**pricerange** *min max*'
  - Alle Fahrzeug(e) mit Preis  $\geq min$  &  $\leq max$  ausgeben  
Anmerkung: *min/max* sind Integer-Werte

Anmerkung: eine Plausibilitätskontrolle für die angegebenen min/max-Werte ist nicht erforderlich.

Beispielaufruf:

```
java FahrzeugClient <Datei> pricerange 50000 100000
```

Ausgabe:

Typ: PKW  
Id: 5  
Marke: Tesla  
Modell: Model S  
Baujahr: 2016  
Grundpreis: 65000.00  
Servicejahr: 2016  
Preis: 55250.00

Typ: BUS  
Id: 9  
Marke: MAN  
Modell: City  
Baujahr: 2018  
Grundpreis: 100000.00  
Servicejahr: 2019  
Preis: 91000.00