

## Aufgabe 9 OCL

### Annahmen:

- es wird = anstatt == verwendet
- mehrere Invarianten oder Conditions werden benannt und jeweils mit inv, pre oder post gekennzeichnet

### Methoden:

```
fuegeZuVerkaufsgruppeHinzu(Verkaufsgruppe vg, Mitarbeiter ma){}
```

Funktion:

einen Mitarbeiter zu einer Verkaufsgruppe hinzufügen

Vorbedingung:

Mitarbeiter existiert | gegeben, da Mitarbeiterobjekt übergeben wird --> automatisch erfüllt

Verkaufsgruppe existiert | gegeben, da Verkaufsgruppenobjekt übergeben wird --> automatisch erfüllt

Mitarbeiter ist nicht in Verkaufsgruppe

Mitarbeiter ist nicht Verkaufsgruppenleiter

Nachbedingung:

Mitarbeiter ist in Verkaufsgruppe

### OCL:

```
context Mitarbeiterverwaltung::fuegeZuVerkaufsgruppeHinzu(vg:
```

```
Verkaufsgruppe, ma: Mitarbeiter)
```

```
pre nichtInVerkaufsgruppe:
```

```
vg.mitglieder->
```

```
forall(mg | mg.id <> ma.id))
```

```
pre nichtLeiter:
```

```
vg.leiter->
```

```
forall(l | l.id <> ma.id )
```

```
post:
```

```
vg.mitglieder->
```

```
includes(ma)
```

```
loescheMitarbeiter(long id){}
```

Funktion:

Löscht einen Mitarbeiter aus dem System

Vorbedingung:

Mitarbeiter existiert

Mitarbeiter ist nicht Mitglied einer Verkaufsgruppe

Mitarbeiter ist nicht Verkaufsgruppenleiter

Nachbedingung:

Mitarbeiter ist weg (keine ID mehr vorhanden)

```
context Mitarbeiterverwaltung::loescheMitarbeiter(id: long)
```

```
pre existiert:
```

```
Mitarbeiter.allInstances->
```

```
exists(ma | ma.id = id)
```

```
pre nichtInVerkaufsgruppe:
```

```

Verkaufsgruppe.allInstances->
  forall(vg | vg.mitglieder->
    forall(mg | mg.id <> id))

pre nichtLeiter:
Verkaufsgruppe.allInstances->
  forall(vg | vg.leiter.id <> id)

post:
Mitarbeiter.allInstances->
  forall(ma | ma.id <> id)

erstelleVerkaufsgruppenleiter(String n, String ad, LocalDate gb){ return
Mitarbeiter}
  Funktion:
    Erstellt einen neuen Mitarbeiter
    Erzeugt eine ID
    Ordnet einen Mitarbeiter einer Verkaufsgruppe zu --> sehr
fraglich, da keine Verkaufsgruppe spezifiziert ist
  Vorbedingung:
    #Annahme: Mitarbeiter dürfen die exakt gleichen Namen, Adressen
und Geburtstagen haben, da sie durch die id unterschieden werden
    gb muss bestimmte grÖÙe haben, da nur Mitarbeiter ab 18 Jahren
eingestellt werden
  Nachbedingung:
    Mitarbeiter existiert
    Mitarbeiter ist Leiter einer Verkaufsgruppe

#Alter wird berechnet <-- unklar wie darzustellen
context Mitarbeiterverwaltung::erstelleVerkaufsgruppenleiter(n: String,
ad: String, gb:LocalDate)
pre: year(currentDate - gb) >= 18

post existiert:
Mitarbeiter.allInstances->
  exists(ma | ma.name = n, ma.adresse = ad, ma.geburtstag = gb)

post istLeiter:
Mitarbeiter.leiterVon <> null

Klasse Mitarbeiter
  Invariante:
    Nicht gleichzeitig Verkaufsgruppenleiter und Mitglied der selben
Gruppe

#Annahme: man kann ganze Objekte einfach vergleichen. Sonst schwierig, da
es keine GruppenIds gibt
context Mitarbeiter inv: leiterVon <> mitgliedVon

```

Klasse Verkaufsgruppe

    Invariante:

        Mitglieder muss  $\geq 5$  sein (Auf Blatt 1 wurde noch gesagt, dass es mindestens ein Mitglied geben muss. Hier sind es 0 bis 5)

        Muss immer einen Leiter haben

context Verkaufsgruppe

    inv maxGroesse: size(mitglieder)  $\leq 5$

    inv hatLeiter: leiter  $\neq$  null