

Formale objektorientierte Software-Entwicklung

Priv.-Doz. Dr. R. Hennicker, Dr. P. Kosiuczenko

Aufgabe 1 Blatt 5

Aus den vorhandenen Assoziationen ergeben sich folgende **implizite Invarianten**.
Aus Multiplizitäten ≥ 1 an 'navigierbaren' Enden:

```
context Flugmanager  
  inv invHatFluglinie : fluege -> forAll(not(fluglinie=nil))
```

Aus bidirektionalen Assoziationen:

```
context Flugmanager  
  inv invBidirekt :  
    Flugmanager.allInstances() -> forAll(fm |  
      fm.fluege -> forAll(f | f.fluglinie.fluege -> includes(f)) and  
      fluglinien -> forAll(fl | fl.fluege -> forAll(f | f.fluglinie = fl)))
```

Aus gerichteten Assoziation:

```
context Flugmanager inv invPartner :  
  Flugmanager.allInstances() -> forAll(fm |  
    fluglinien -> forAll(fl | fl.partner -> forAll(pl | pl.partner->includes(fl))))
```

Aus Multiplizitäten an 'nicht-navigierbaren' Enden:

```
context Flugmanager  
  inv invPilotGehoertzuFluglinie :  
    Flugmanager.allInstances() -> forAll(fm |  
      fm.piloten -> forAll(p |  
        (fluglinien -> select(fl | fl.personal -> includes(p))) -> size() <= 1))
```

```
context Flugmanager  
  inv invFlugGehoertzuPilot :  
    let anzPiloten(f:Flug):Integer =  
      (piloten -> select(p | p.einsatz -> includes(f))) -> size()  
  in  
    Flugmanager.allInstances() -> forAll(fm |  
      fluege -> forAll(f | anzPiloten(f) >= 1 and anzPiloten(f) <= 3))
```

...ähnlich für *Fluglinie* und *Flugmanager*, *Flug* und *Flugmanager*, *Pilot* und *Flugmanager*.