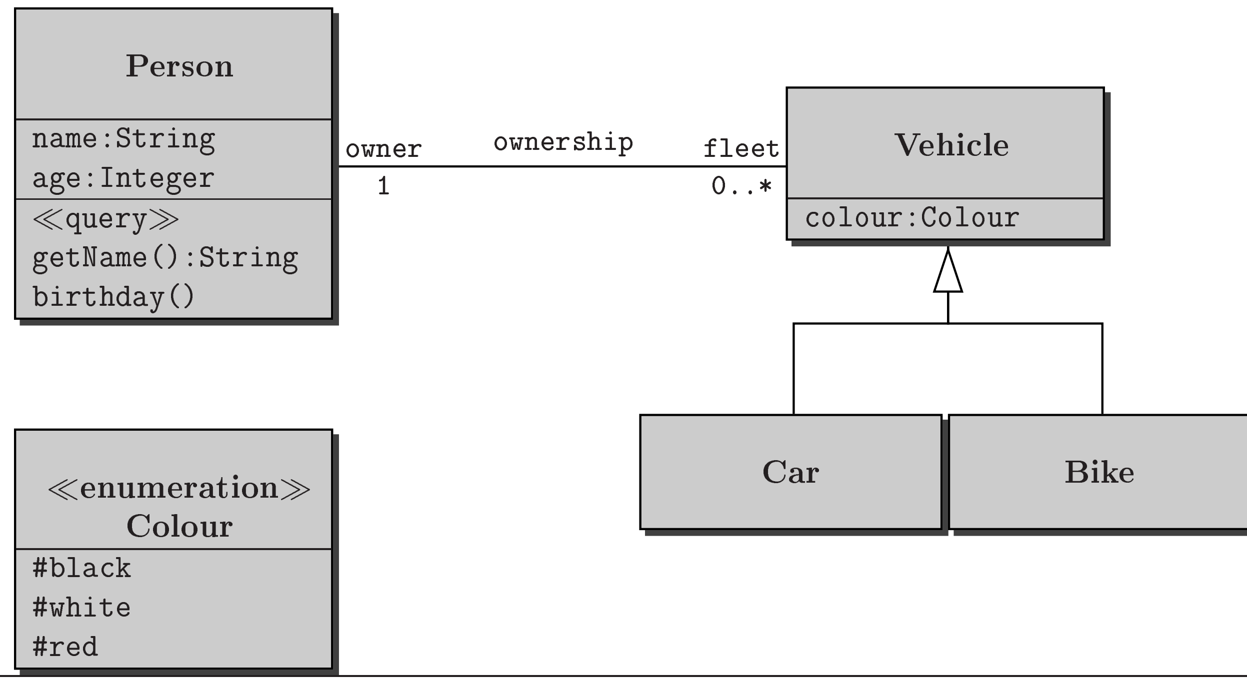
Einführung in UML/OCL

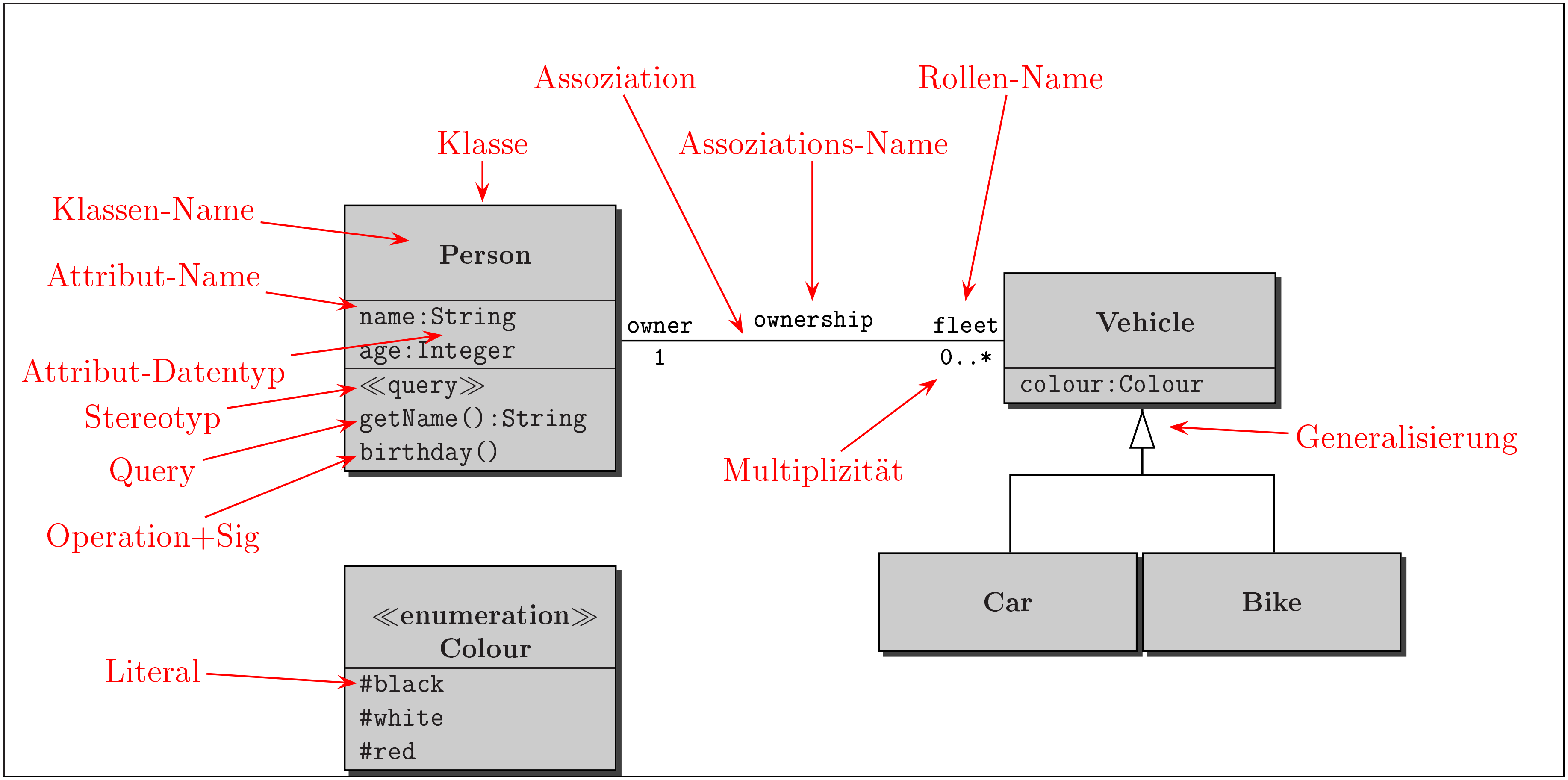
Bernhard Beckert

UNIVERSITÄT KOBLENZ-LANDAU

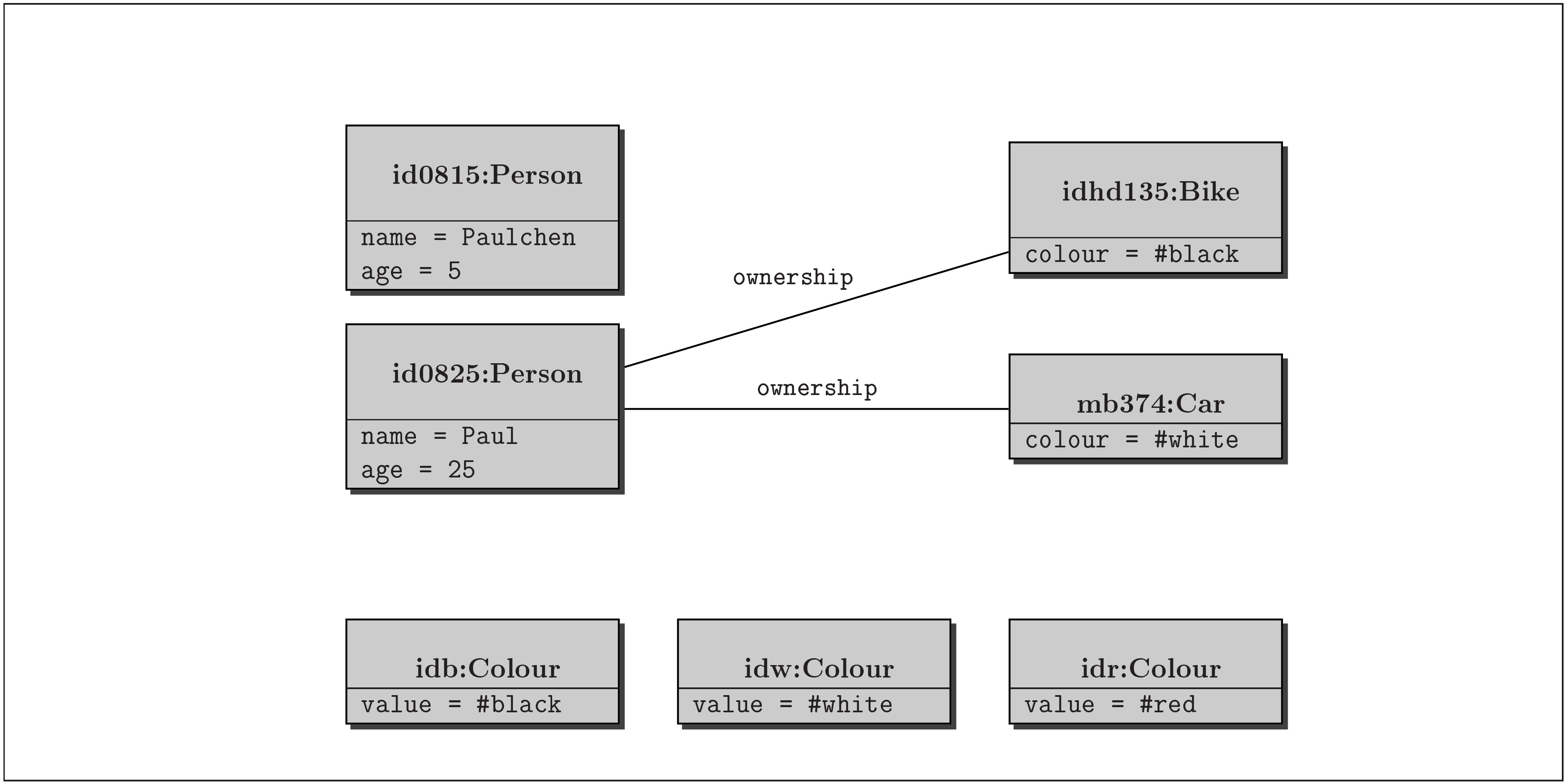
Klassendiagramme – Beispiel



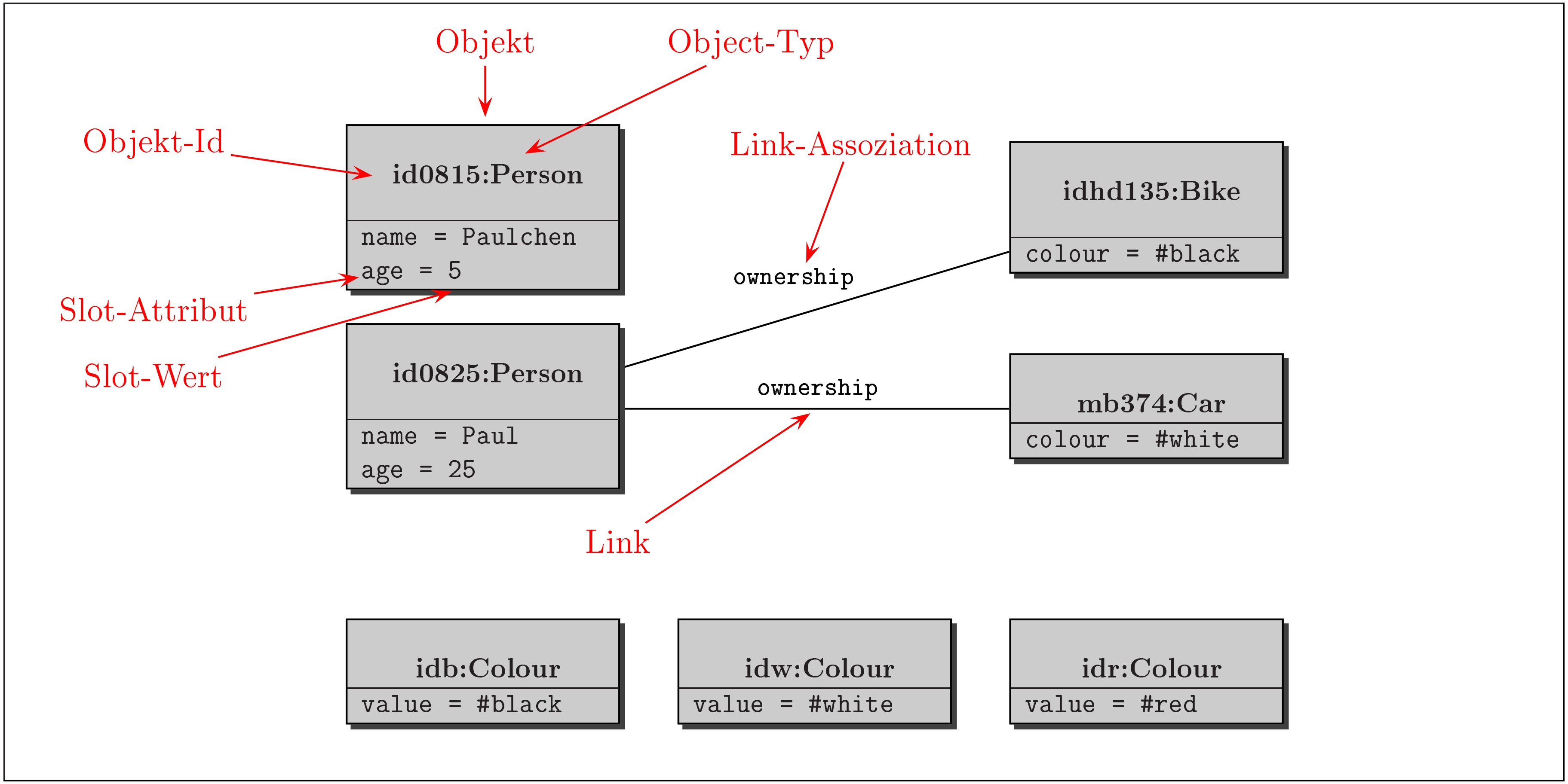
Klassendiagramme – Syntax



Objektdiagramme – Beispiel



Objektdiagramme – Syntax



Systemzustand

- **gewähltes Objektdiagramm (identifizierbare Objekte, Typzuordnung, Attribut-Wert-Zuordnung, Links)**
- **(Standard)-Interpretation der vordefinierten Datentypen für String, Integer, Real**
- **gewählte Interpretation für Queries**

  $\mathcal{I}(getName) : \mathcal{I}(Person) \rightarrow \mathcal{I}(String)$

Systemrealisierung

$[State, rel]$

wobei

$rel \subset Operation \times State \times State$

Ziel:

Ausschließung unerwünschter Systemrealisierungen (ähnlich Multiplizitätsangabe)

Mittel:

Formale (d.h. formale Grammatik) Sprache zur Formulierung von Eigenschaften von Modellelementen

Syntax:

Anlehnung an Java, Typsystem

OCL – Beispiel Navigation

Personen sollen immer volljährig sein

```
context Person inv:
self.age >= 18
```

`self` **ist vordefinierte Variable, hier vom Typ** `Person`

Typregeln für Ausdrücke sind analog denen für Java

`>=` **ist deklariert in Bibliothek für** `Real`

Fahrzeugbesitzer sollen immer volljährig sein

```
context Vehicle inv:
self.owner.age >= 18
```

OCL – Beispiel Collection (Set, Bag, Sequence)

Keiner hat mehr als 3 Fahrzeuge

```
context Person inv:
self.fleet->size <= 3
```

`size` **ist deklariert in Bibliothek für** `Collection`

```
self.fleet->forAll(v| v.colour = #black)
```

Aufgabe:

```
self.fleet->select(v| v.colour = #black)->size >= 1
```

OCL – iterate

context Person **inv:**

```
self.fleet->iterate(v; acc : Integer = 0 |
if (v.colour = #black) then acc + 1 else acc)) >= 3
```

Aufgabe: Simulation von forAll, exists, select, collect mittels iterate.

OCL – Beschreibung von Queries

```
context Person::getName():String post:

result = self.name
```

BESSER als Invariante:

```
context Person inv:

self.getName() = self.name
```

OCL – Vor-/Nachbedingung für Operationen

Beschreibung von Zustandsübergang

context `Person::birthday():Integer` **post:**

```
self.age = self.age@pre + 1
```