Objektorientierte Programmierung: Relationen

13.05.15

Constructors

How are objects created from classes?

Each object is basically a variable (of a defined class)

Must be initialized to some correct initial value

This is done by a special function: The constructor

- Has the same name as the class
- Is always called in combination with new

Constructors

```
1
2  Car(color, wheels, engine, gasTank)
3  {
4     Color = color;
5     Wheels = wheels;
6     Engine = engine;
7     GasTank = gasTank;
8  }
9
```

Constructors

```
2 Car MyOldtimer = new Car(red, 4, “Otto motor”, 50);
```

MyOldtimer: Car
Color = red
Wheels = 4
Engine = Otto Motor
Gas tank = 50 l

Constructors

```
Car(color, engine, gasTank)
{
    Color = color;
    Wheels = 4;
    Engine = engine;
    GasTank = gasTank;
}
```

Constructors

All information necessary to initialize the attributes of an object is passed by parameters

Some attributes may have default values. These may be omitted in the parameter list of the constructor

In the constructor, the omitted attributes are set to some default value (e.g. a default value for the number of wheels could be 4. But what would be a good default color?)

Destructors

Some languages like PHP, C++, C# (but not ActionScript, Javascript, ...) provide destructors.

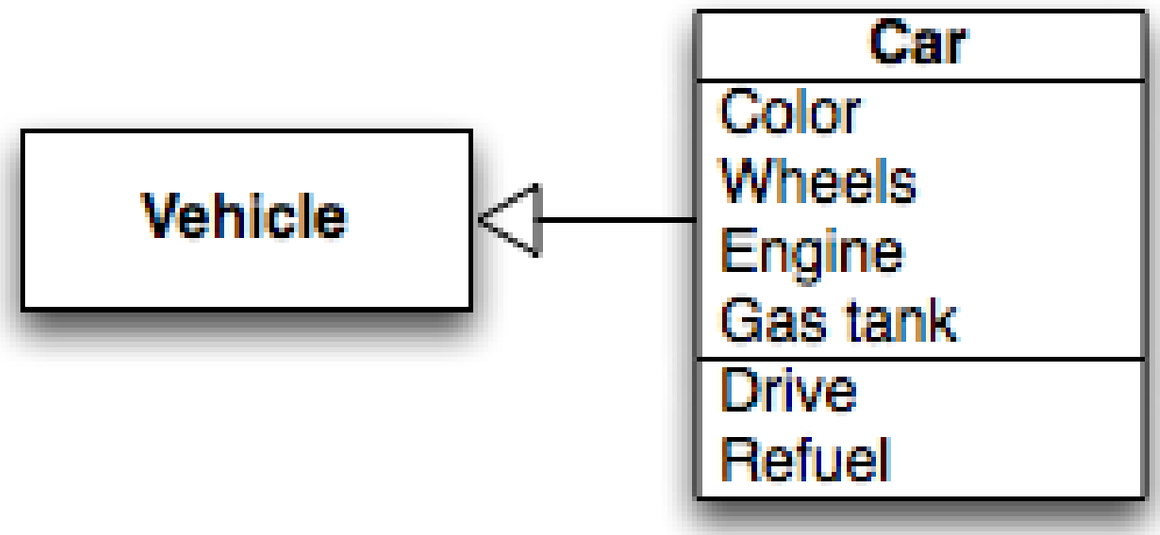
Destructor: A method called when your object is destroyed

Often not necessary: Memory occupied by your object is just freed

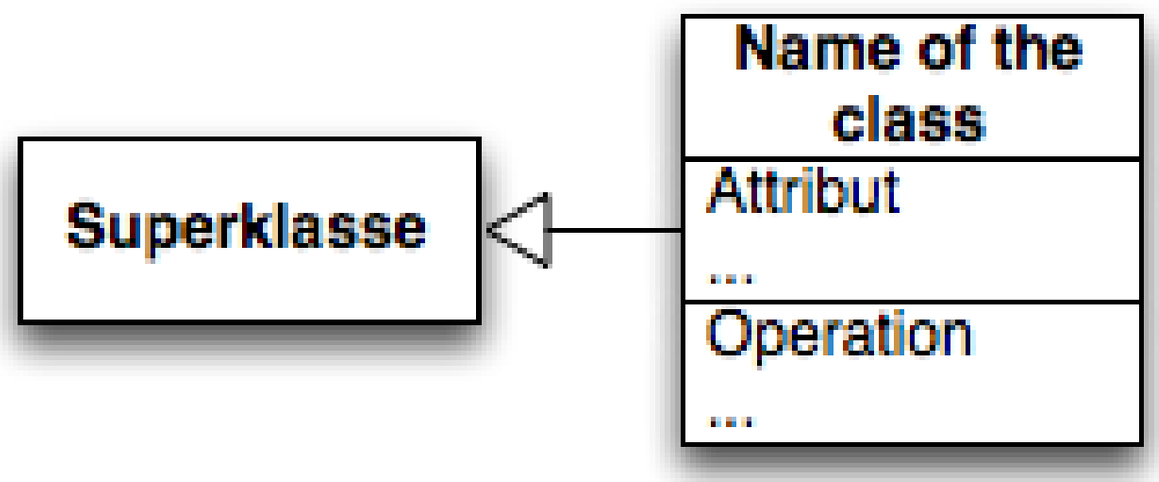
But sometimes you need to do clean up work: Finish work, save data, close files

Vererbung (extends)

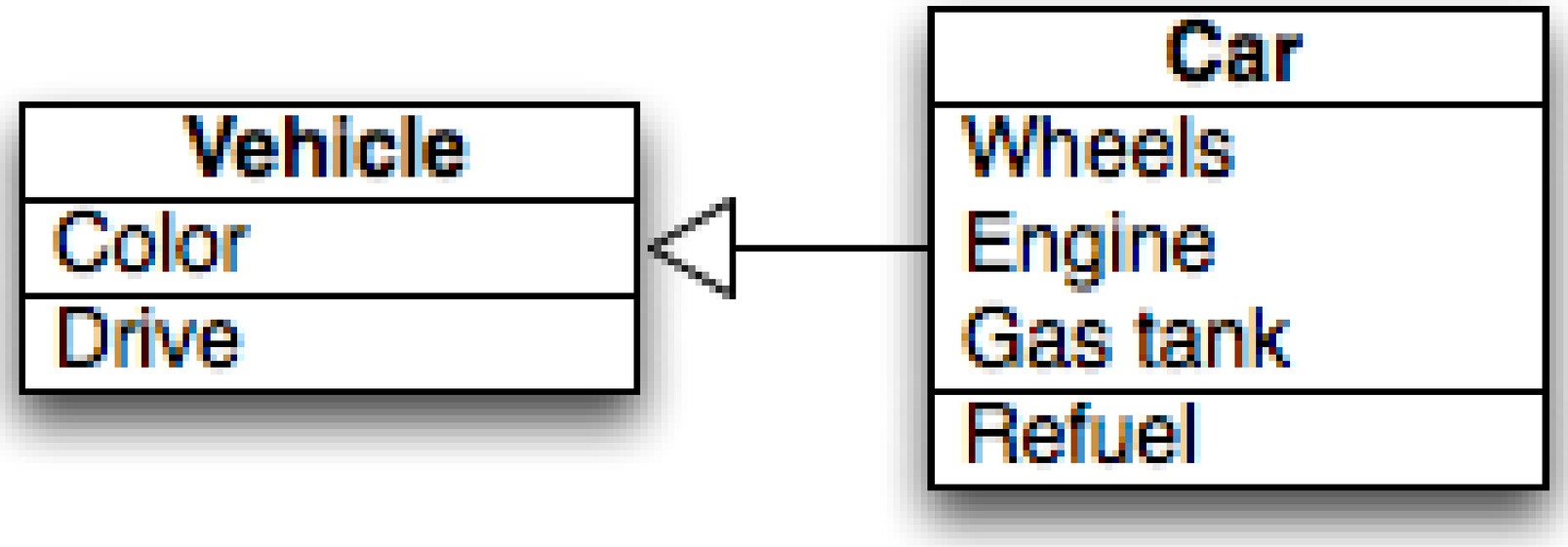
- **Die “ist ein”- Beziehung, nennt man “sub-classing” / Vererbung**
- **“Car” ist eine sub-class von “Vehicle”**



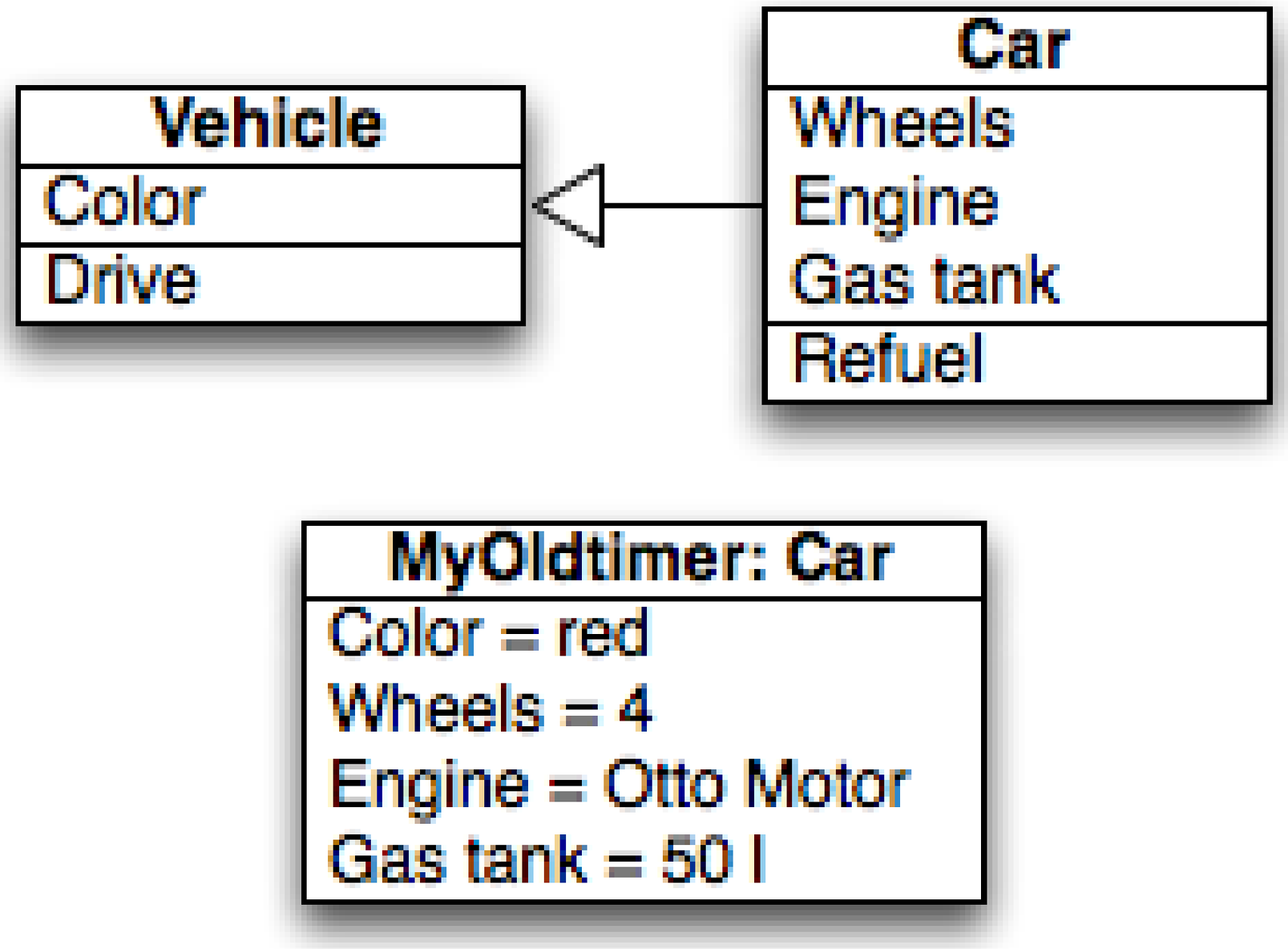

```
1  class OwiKaefer1 extends Kaefer {
2
3    float noiseP;
4    PVector target;
5    boolean hasTarget;
6    int targetMemory;
7
8    color obstacle;
9
10   OwiKaefer1() {
```



Vererbung (extends)

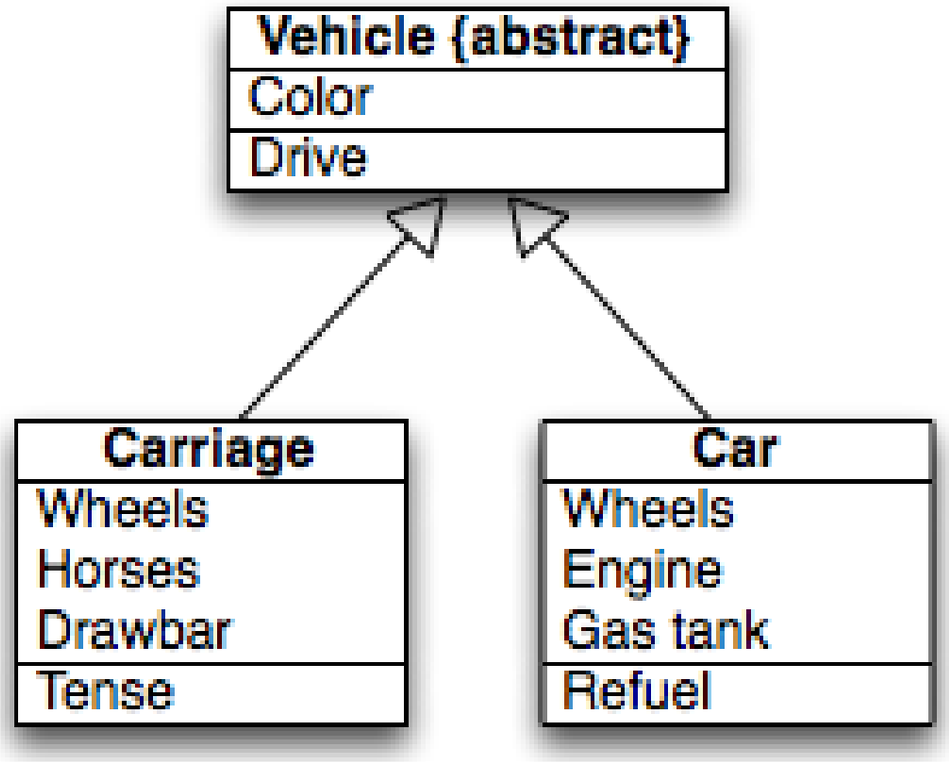


Vererbung (extends)



Hierarchie

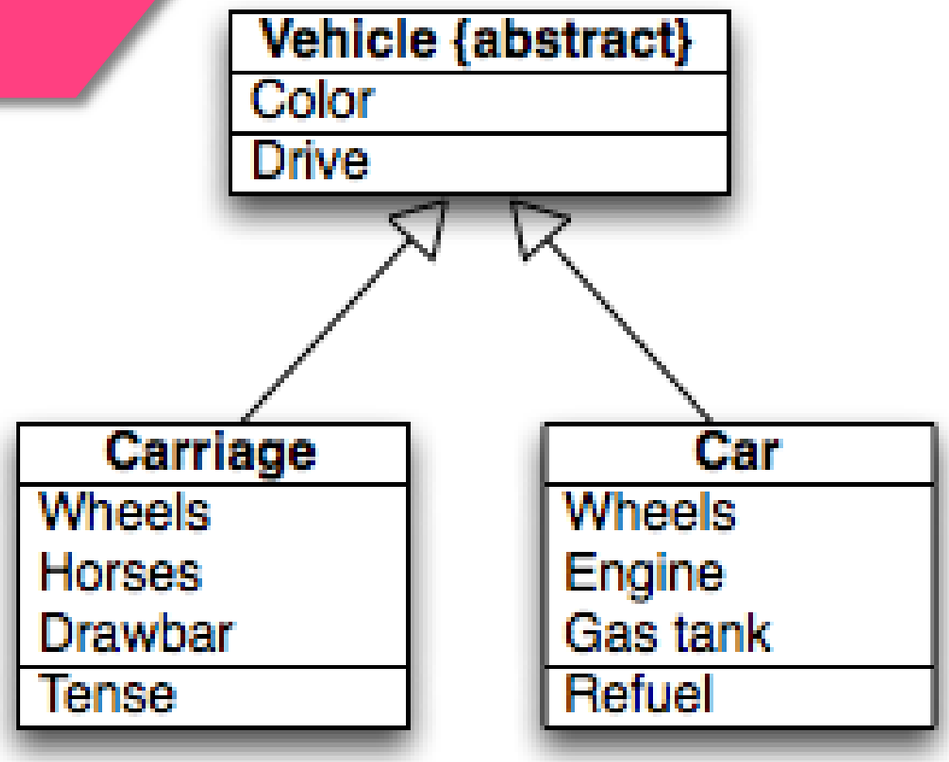
- Another term for super class is base class or parent class
- A sub-class is often called child class
- Creating a sub-class from a super-class is called derivation (verb: to derive = germ.: Ableitung), or specialization



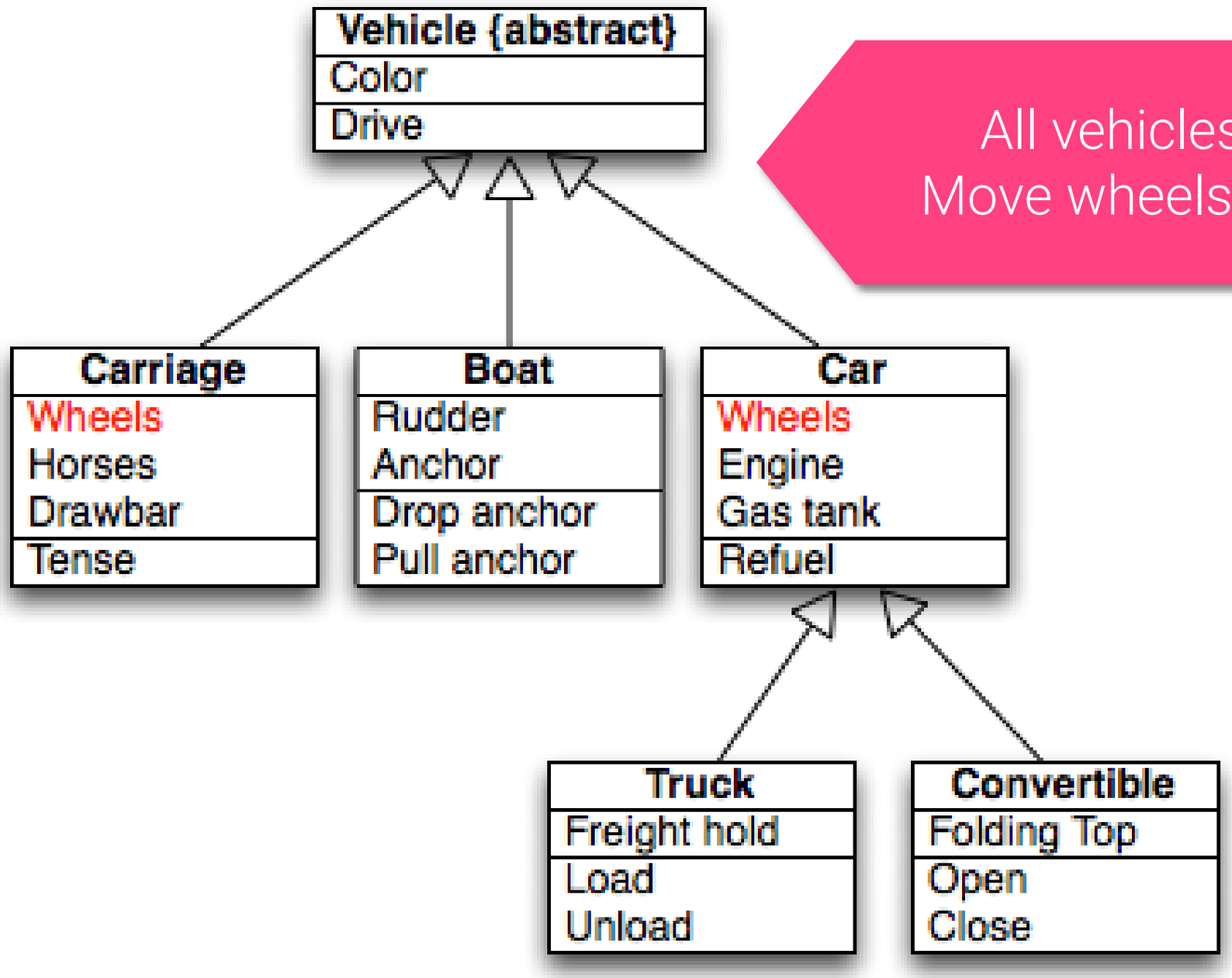
Abstrakte Klasse

There is no “pure vehicle”

- Some classes are not meant to be instantiated (no objects of that class can exist). Such a class is called abstract class
- Abstract classes represent abstract concepts rather than sets of real objects
- Abstract classes can only serve as base class



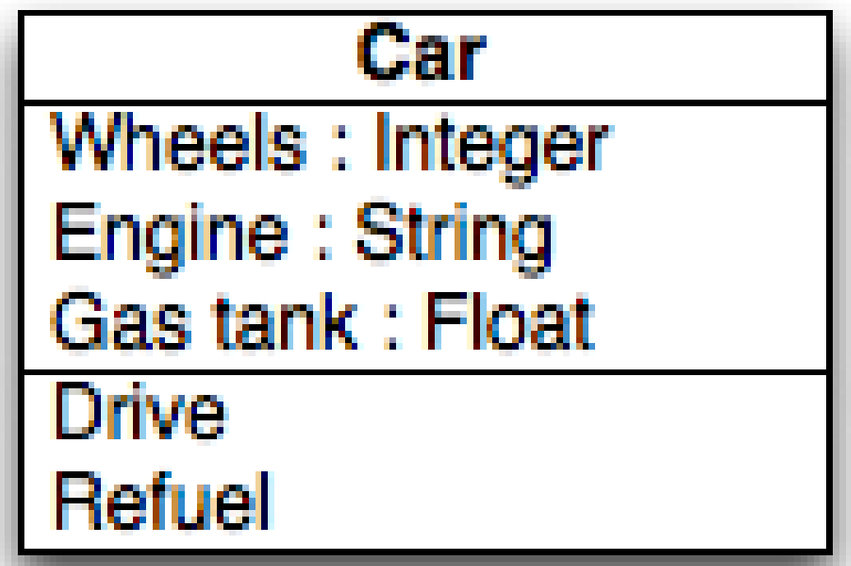
Hierarchie



All vehicles have wheels?
Move wheels to vehicle class?

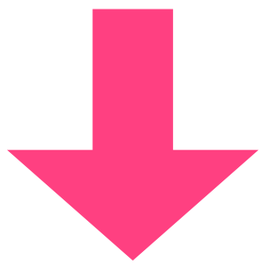
Attribute und Relationen

- Attributes are defined in a class, exist only in the object, contain variables of a certain type
(Number, Boolean, String, …)



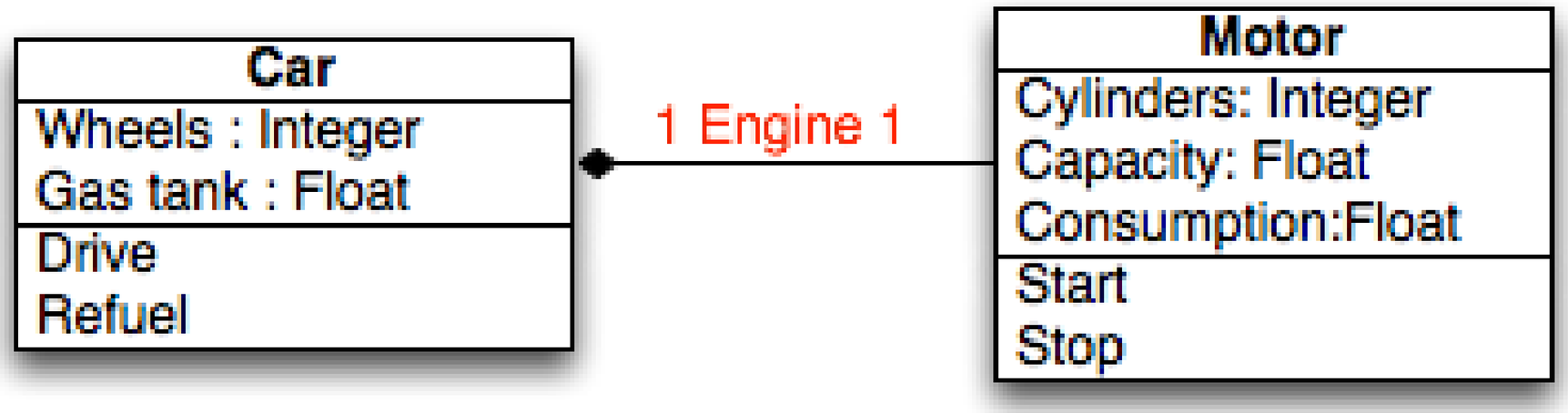
Attribute und Relationen

But in many cases attributes are not simple data like numbers or text but can be real things themselves (like parts of a car)



Real things? Ok, they must be objects!

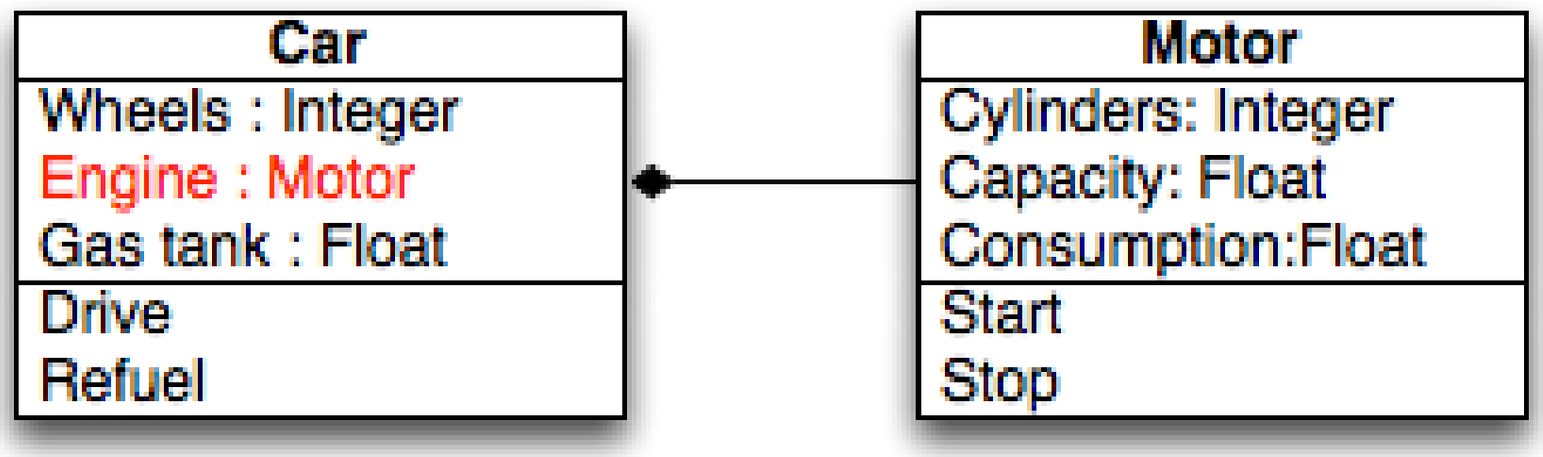
Attribute und Relationen



Numbers on both sides define numbers of objects composed

Komposition

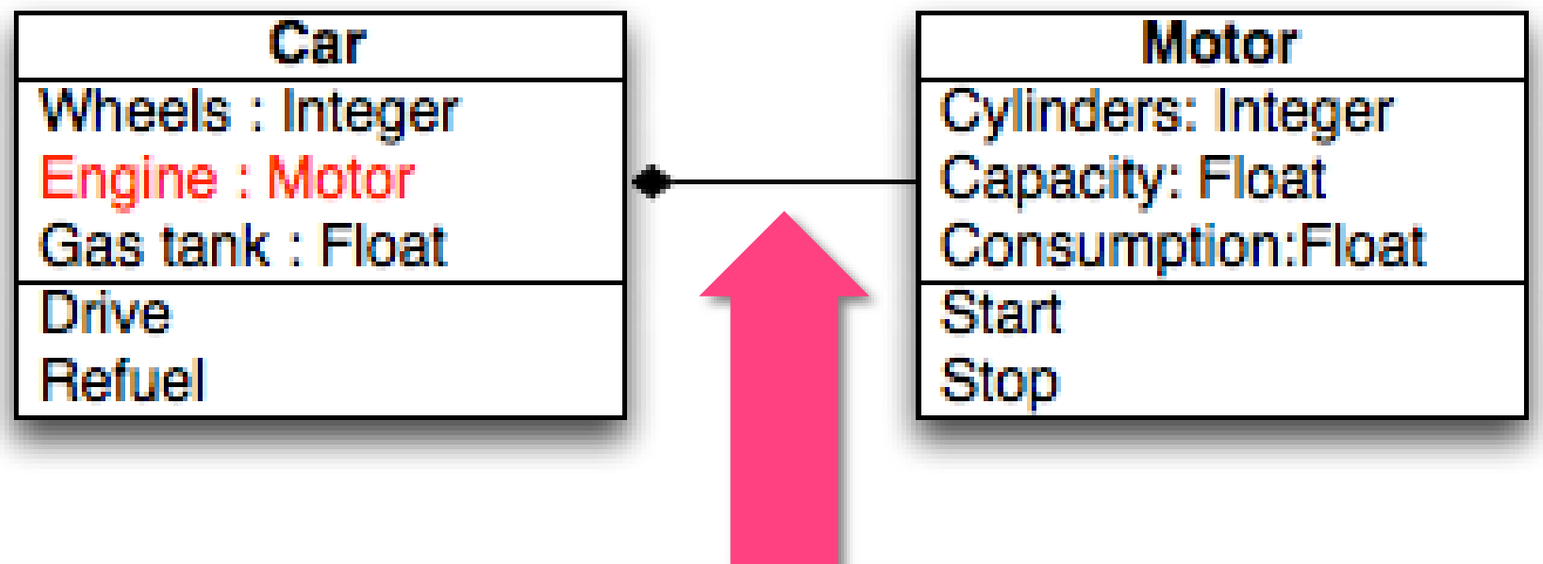
This relation between two classes is called composition



Composition means: The component is mandatory (No cars without engines)!

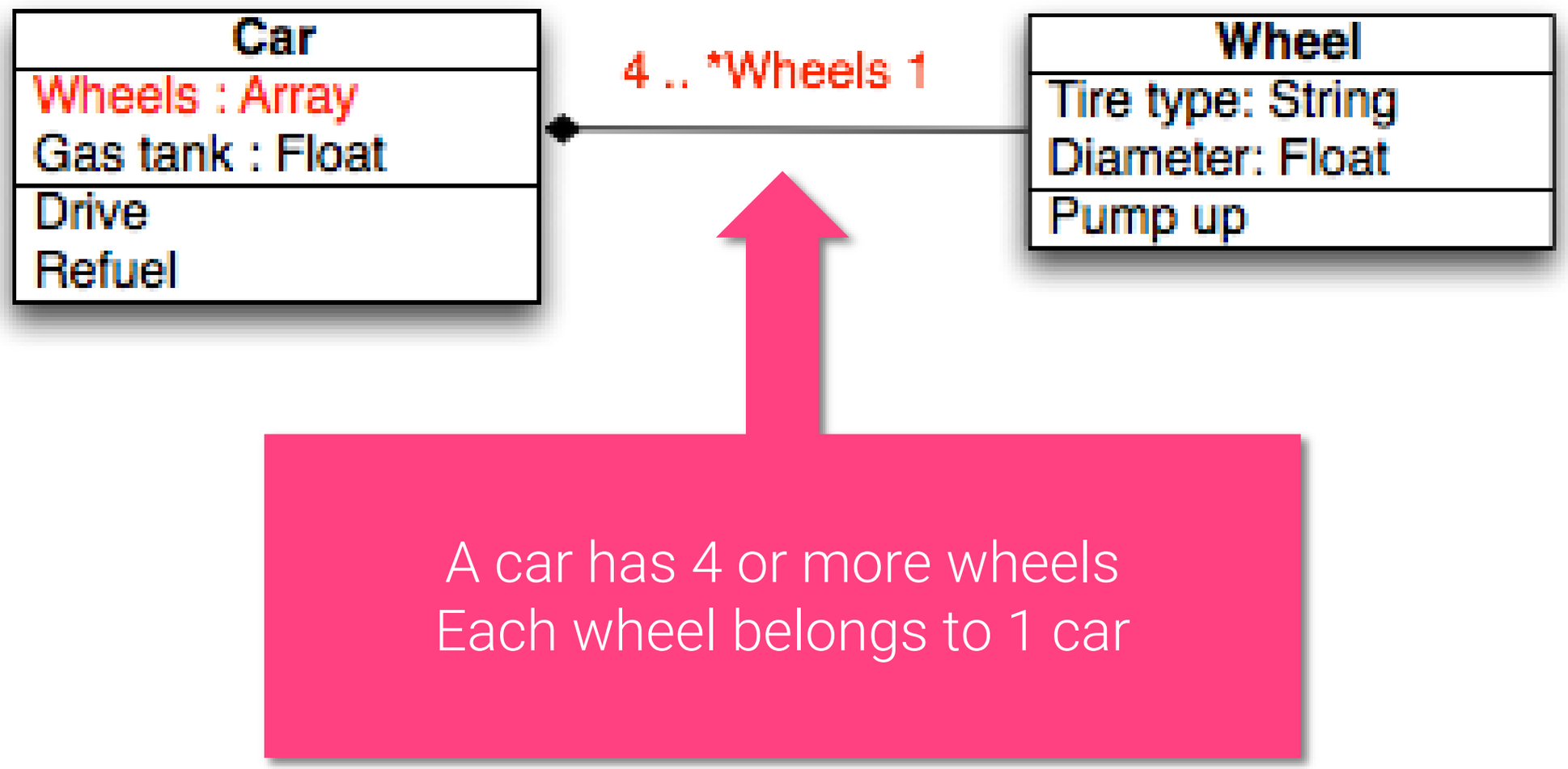
Komposition

This relation between two classes is called composition

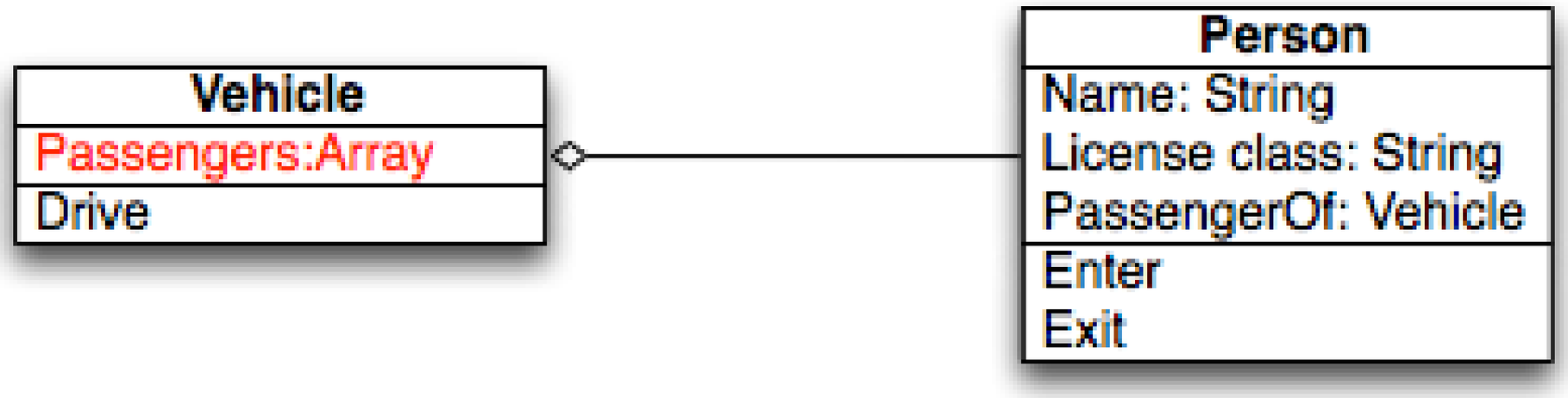


Composition is expressed by a line with a filled diamond on the side of the containing class
The attribute can be omitted

Komposition

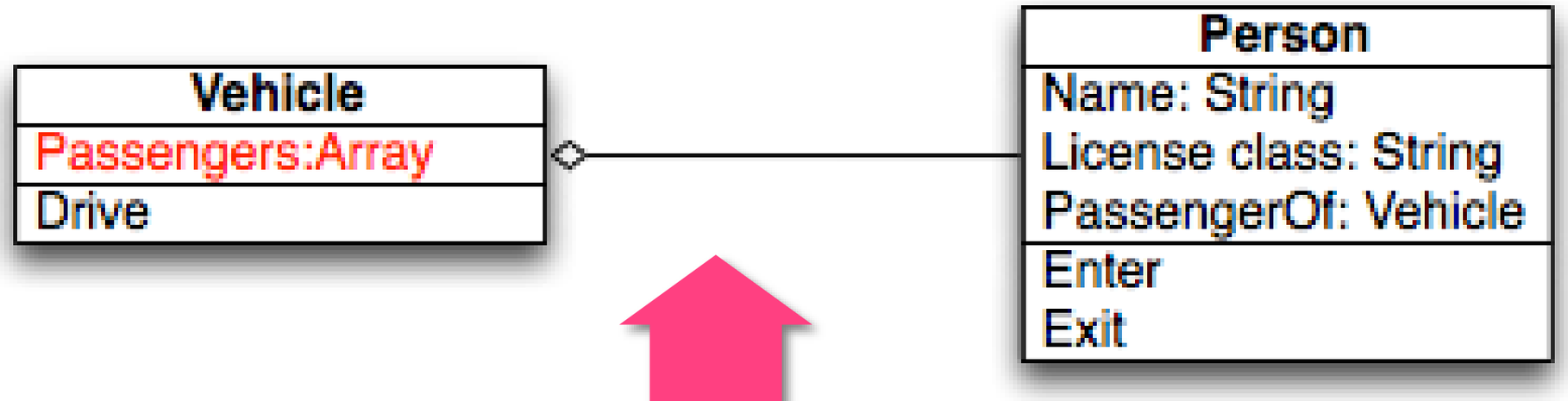


Aggregation



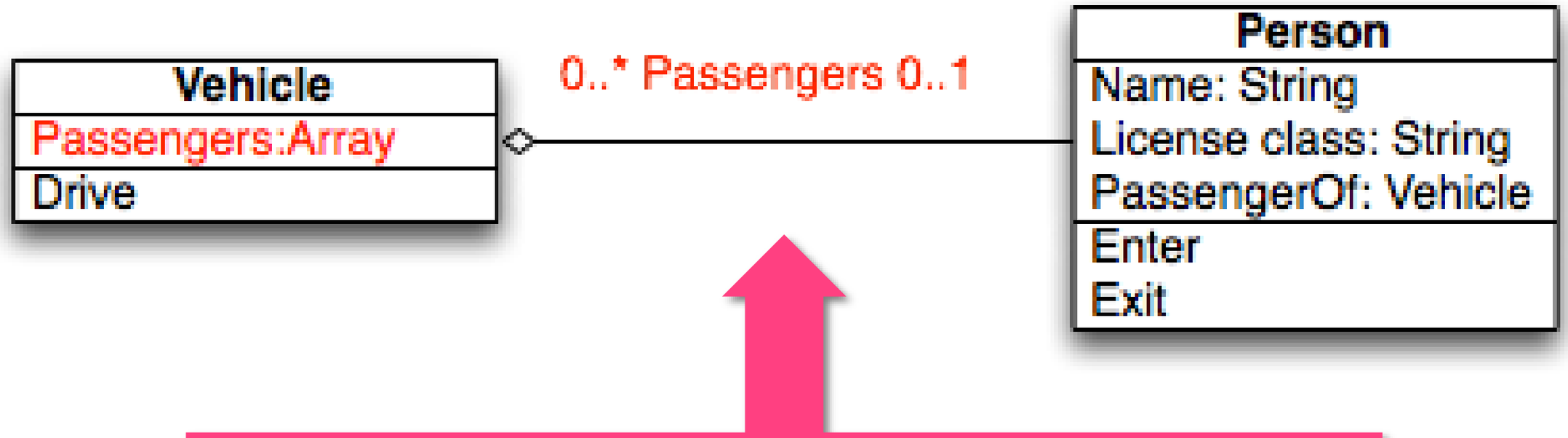
- Aggregation is the softer form of composition
- The aggregated objects may not always exist
- A vehicle (and therefore cars) can exist without passengers

Aggregation



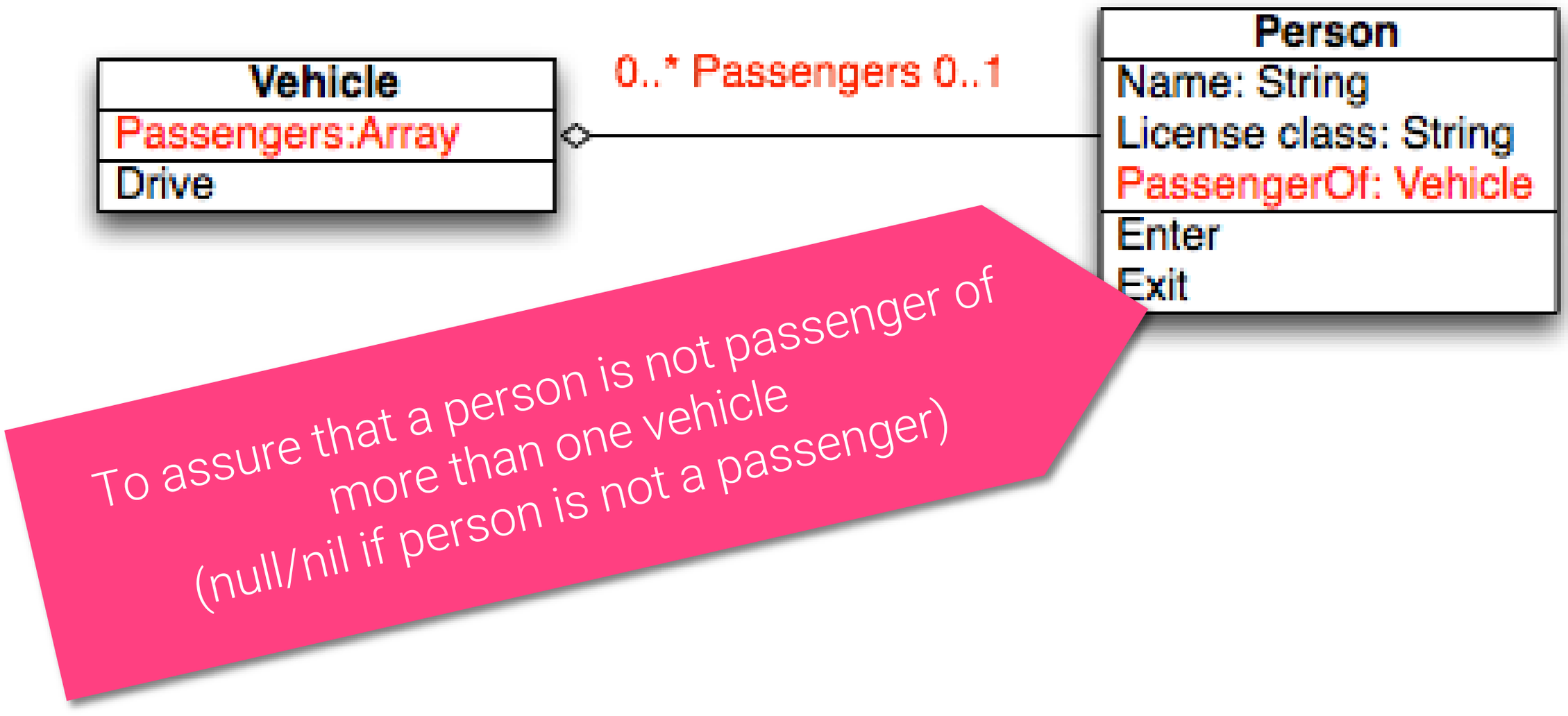
Aggregation is expressed by a line with an empty diamond on the side of the containing class

Aggregation



A vehicle can have an arbitrary number of passengers
A person can only be passenger of 1 vehicle or none at all

Aggregation



Relationen zwischen Klassen

Classes typically form a class hierarchy

Base classes define the basic abstract concepts. When a class is abstract, no objects of it can be created

Derived classes are more specialized, describe sets of real things and inherit methods and attributes from base classes

Composition means: An object of one class can only exist with objects of another (e.g. parts of a car)

Aggregation means: An object of one class typically contains objects of other classes (cars and passengers)

Unified Modeling Language

- Software development is an industry.
- Needs industrial, standardized methods for software design
- Object-oriented software is state of the art
- As a developer, artist, business-person you are part of the development process
- You need to speak and understand the language of your team!

Ihre Software muss in UML dokumentiert werden!

Beispiel UML