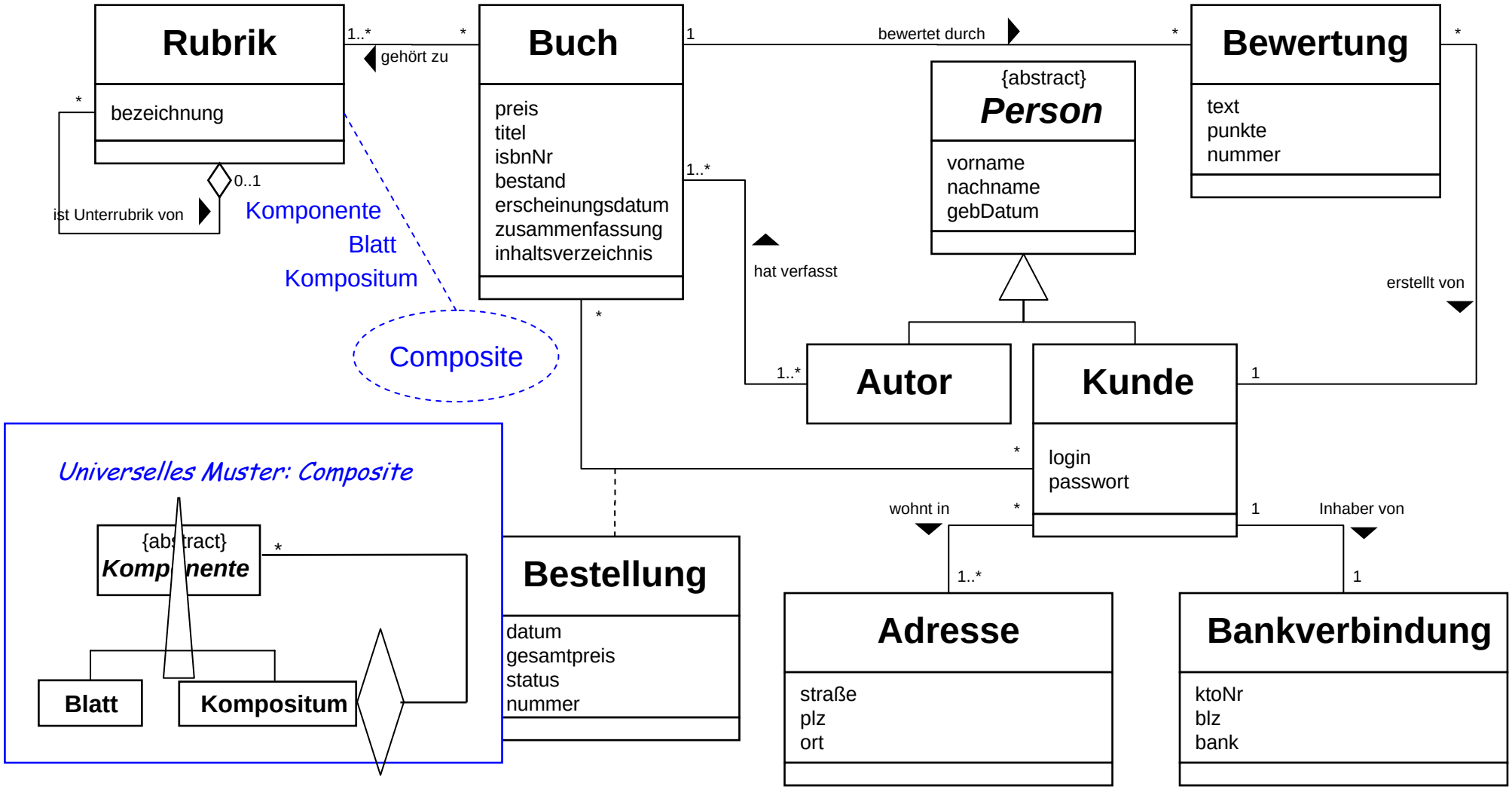
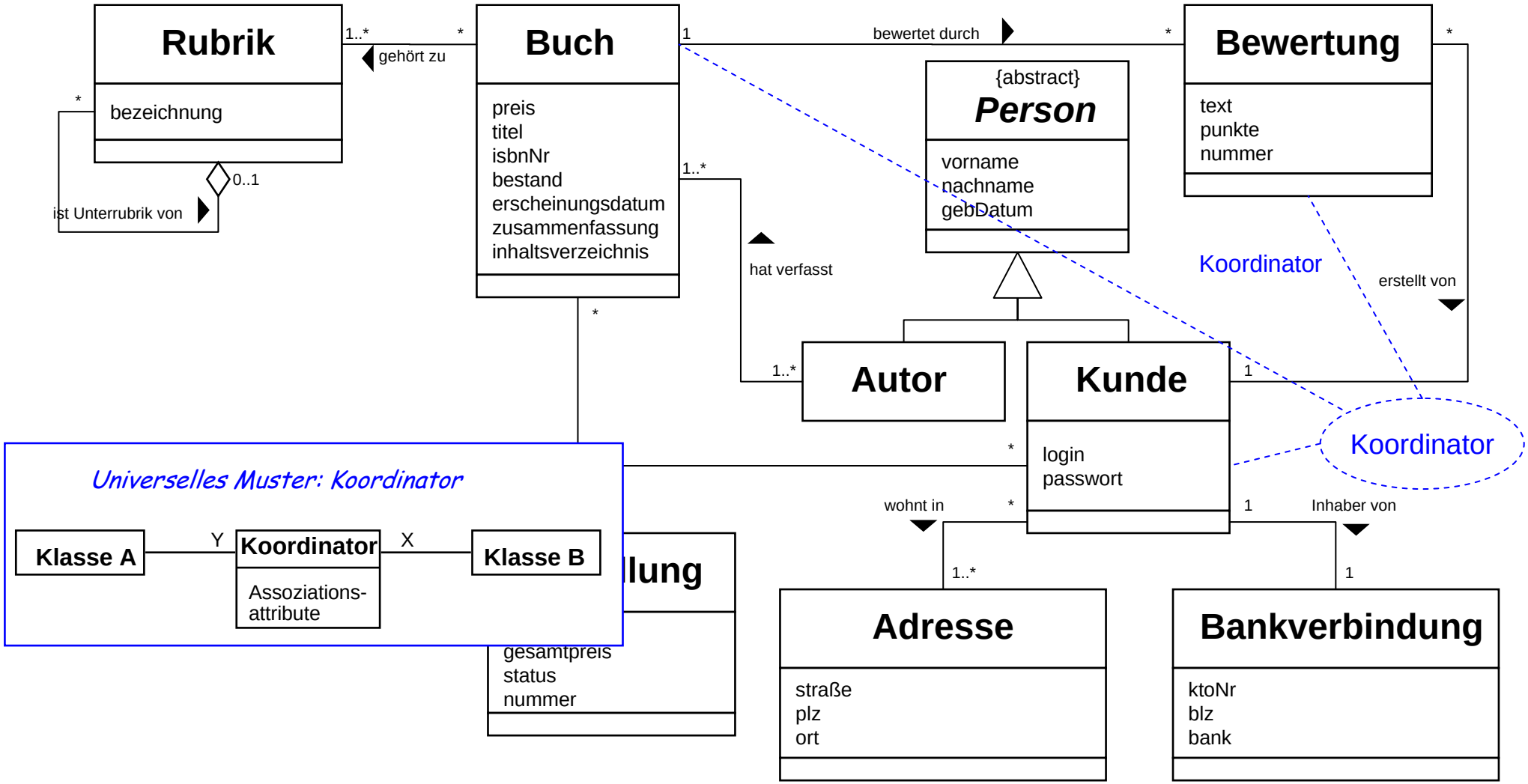
Übung
„Einführung in die Softwaretechnik“

Lösungshinweise zum Übungsblatt 11

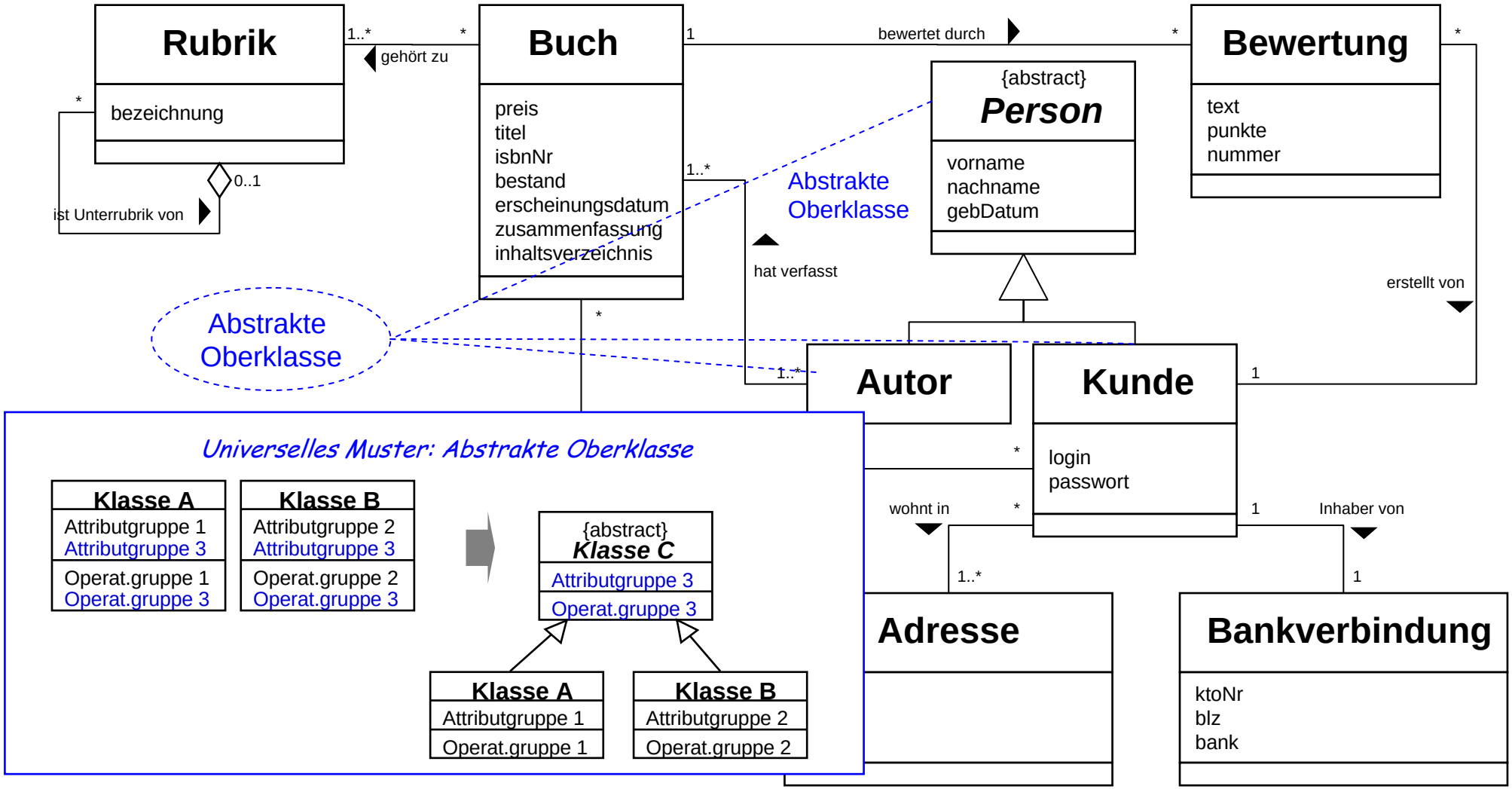
Aufgabe 25a (1 von …): Composite



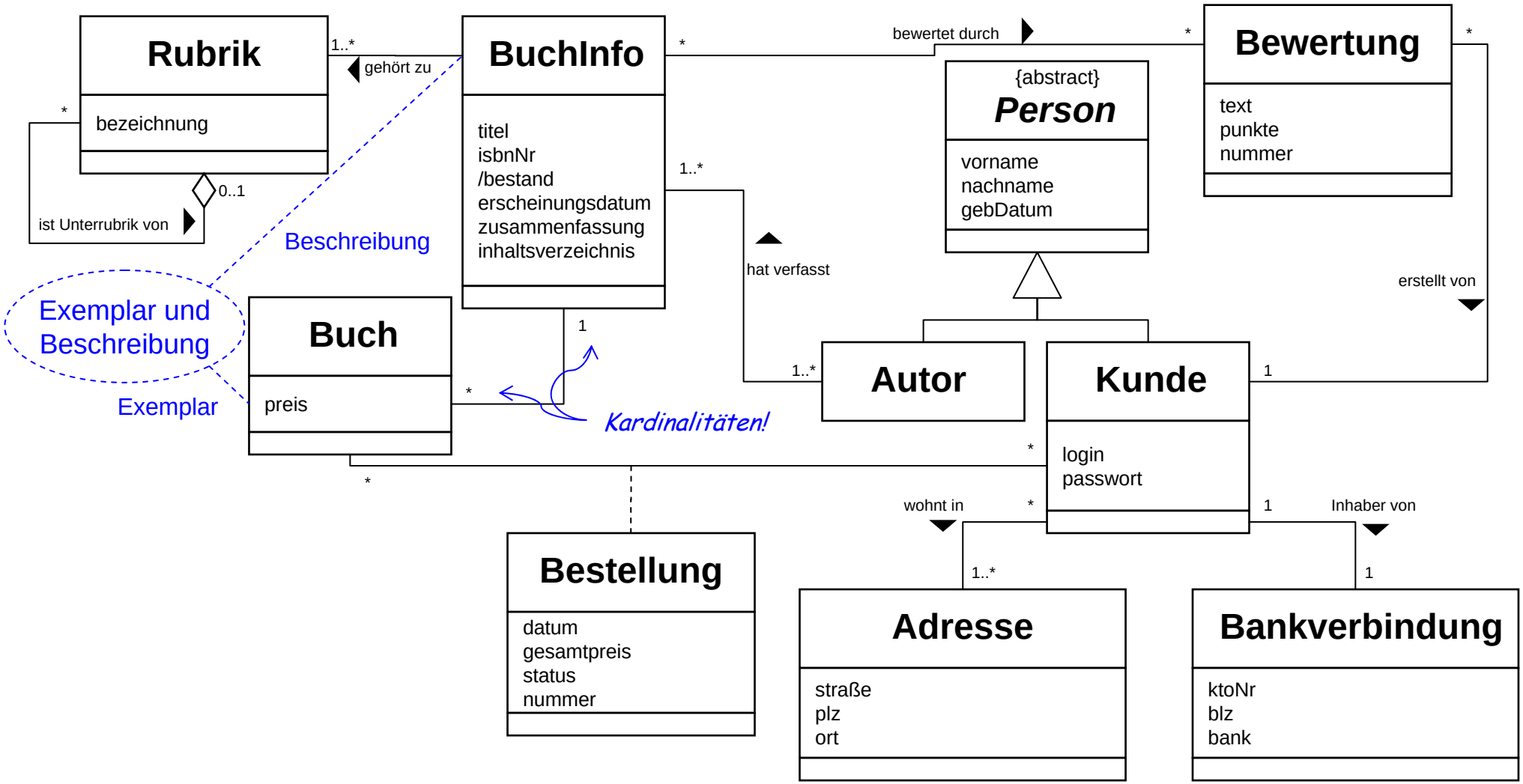
Aufgabe 25a (2 von …): Koordinator



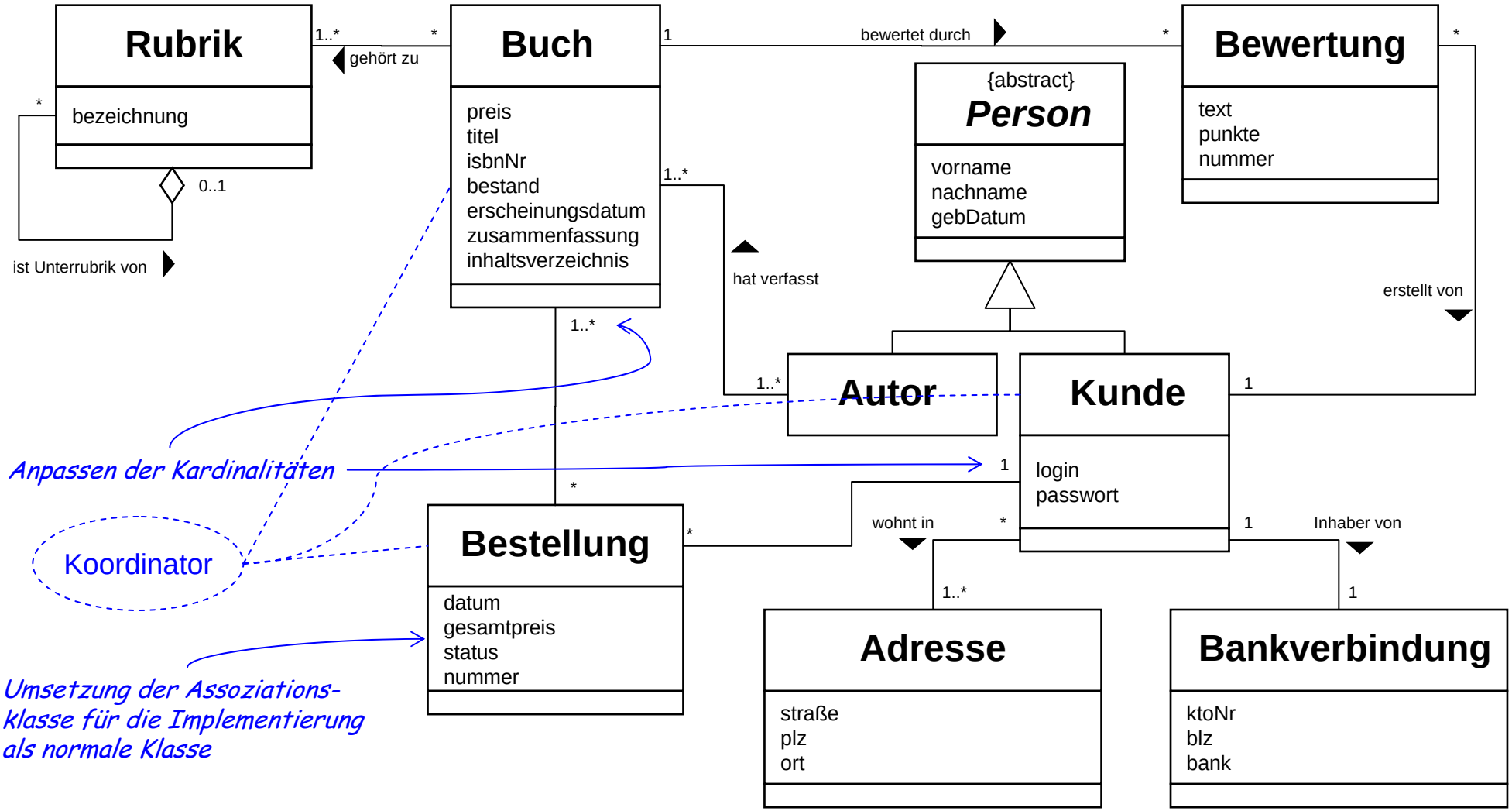
Aufgabe 25a (3 von …): Abstrakte Oberklasse



Aufgabe 25b: Exemplar und Beschreibung



Aufgabe 25c: Koordinator



Aufgabe 26

type generator ARRAY[ITEM]
functions
create: INT × INT → ARRAY
assign: ARRAY × INT × ITEM → ARRAY
first: ARRAY → INT
last: ARRAY → INT
read: ARRAY × INT → ITEM
axioms for a: ARRAY; x,y,n,m: INT; v: ITEM let

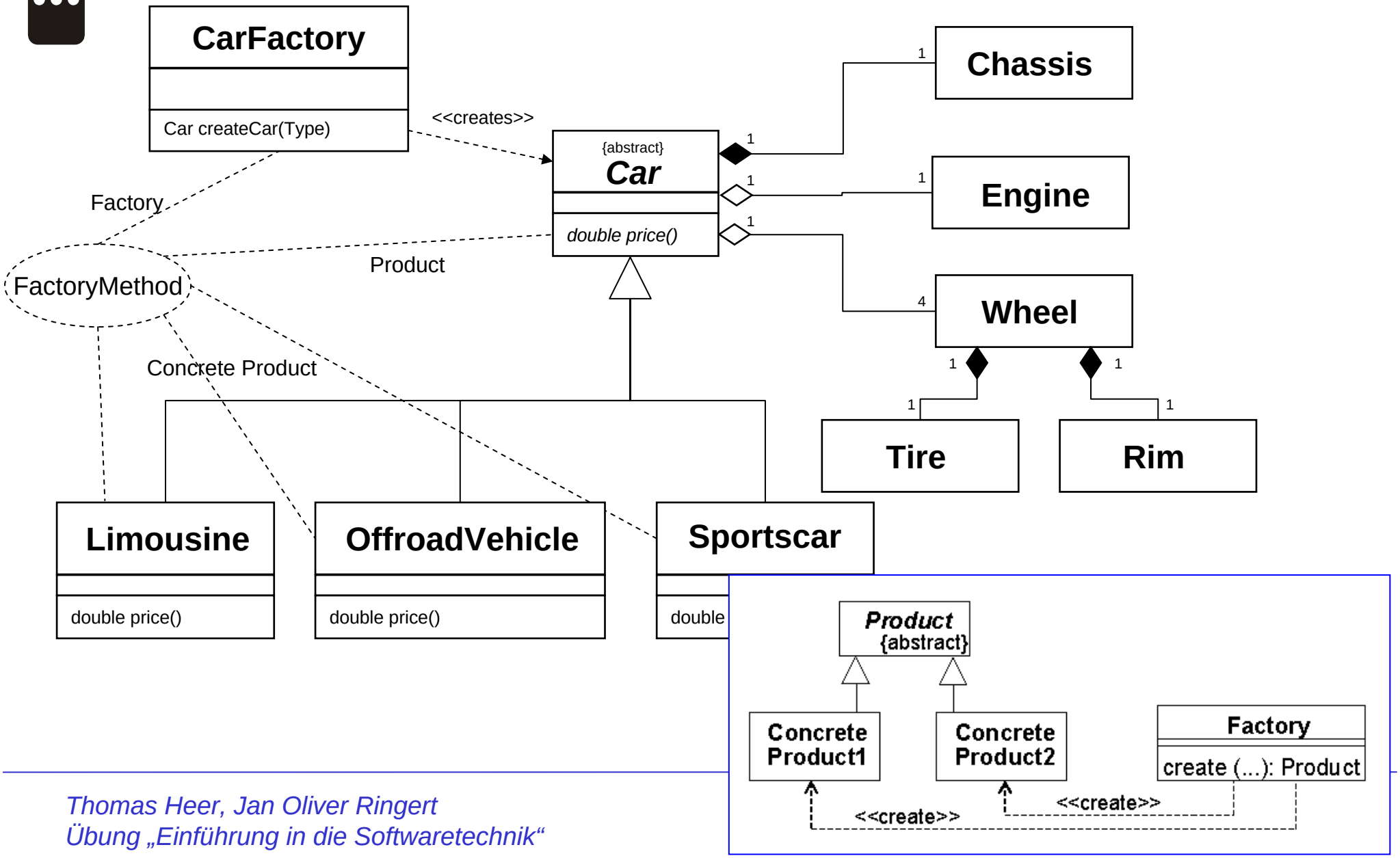
first(create(x, y))	=	x
first(assign(a, n, v))	=	first(a)
last(create(x, y))	=	y
last(assign(a, n, v))	=	last(a)
read(create(x, y), n)	=	undefined
read(assign(a, n, v), m)	=	if m = n then v else read(a, m)

restrictions

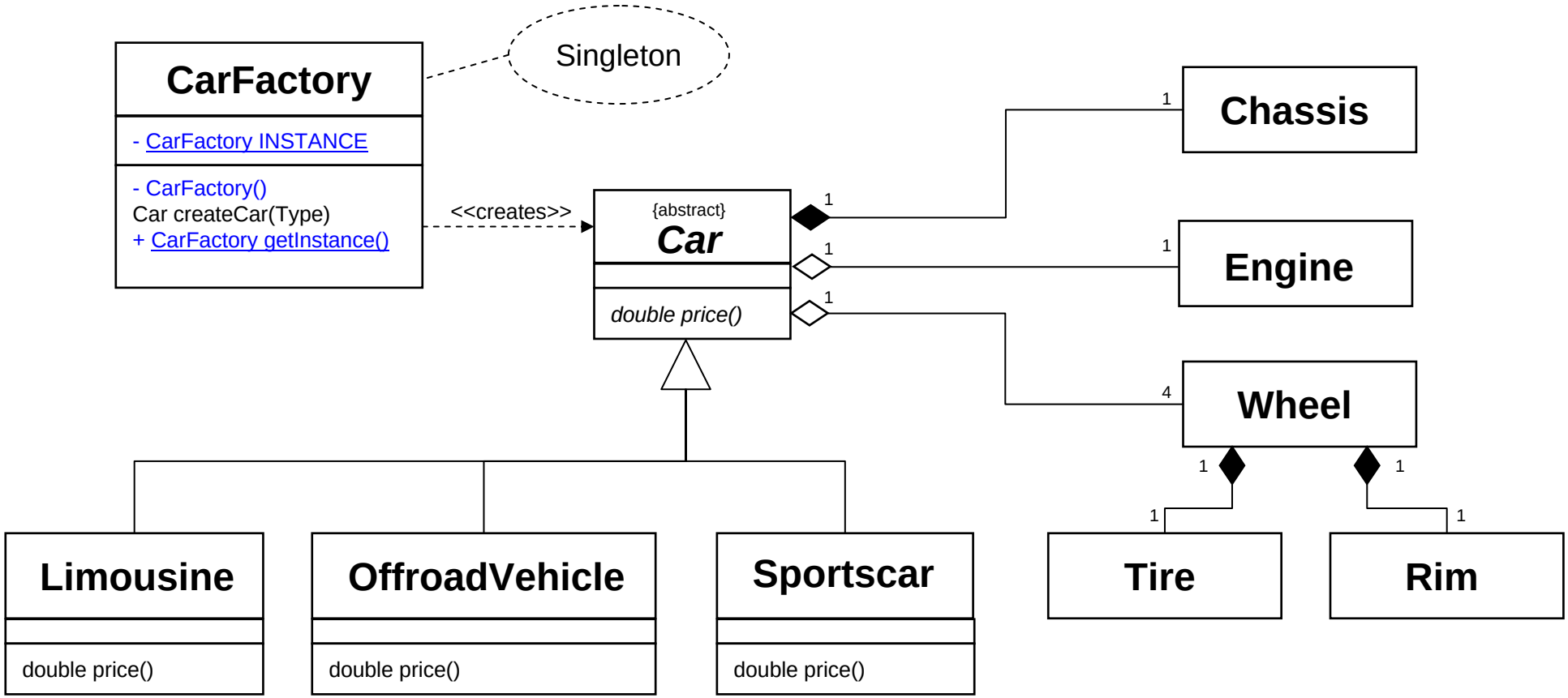
n < first(a) or n > last(a)	=>	read(a, n) = error1
x > y	=>	create(x,y) = error2
n < first(a) or n > last(a)	=>	assign(a, n, v) = error3

end type generator ARRAY

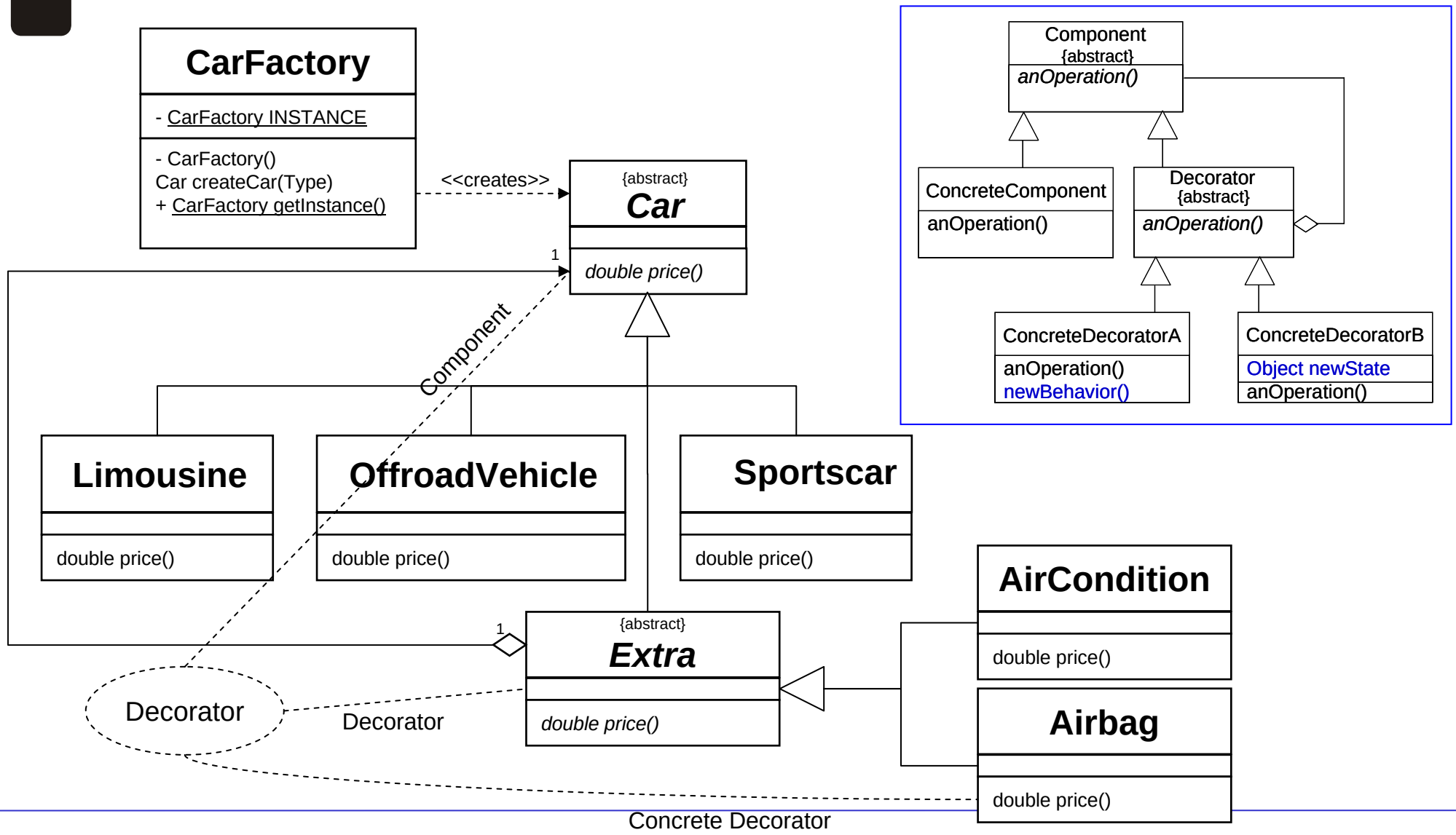
Aufgabe 27a


CarFactory
Car createCar(Type)
<<creates>>
{abstract}
Car
double price()
Chassis
Engine
Wheel
Tire
Rim
1
4
Factory
Product
FactoryMethod
Concrete Product
Limousine
double price()
OffroadVehicle
double price()
Sportscar
double
Product
{abstract}
Concrete Product1
Concrete Product2
Factory
create (...): Product
<<create>>
<<create>>

Aufgabe 27b


CarFactory
- CarFactory INSTANCE
- CarFactory()
Car createCar(Type)
+ CarFactory getInstance()
Singleton
<<creates>>
{abstract}
Car
double price()
Chassis
Engine
Wheel
Tire
Rim
Limousine
OffroadVehicle
Sportscar
double price()
double price()
double price()
1
1
1
1
1
4
1
1
1
1

Aufgabe 27c


CarFactory
- CarFactory INSTANCE
- CarFactory()
Car createCar(Type)
+ CarFactory getInstance()
<<creates>>
{abstract}
Car
double price()
1
Component
Limousine
double price()
OffroadVehicle
double price()
Sportscar
double price()
{abstract}
Extra
double price()
1
Decorator
Decorator
AirCondition
double price()
Airbag
double price()
Concrete Decorator
Component
{abstract}
anOperation()
ConcreteComponent
anOperation()
Decorator
{abstract}
anOperation()
ConcreteDecoratorA
anOperation()
newBehavior()
ConcreteDecoratorB
Object newState
anOperation()

Aufgabe 27d

Car.java

```
public abstract class Car {
    public abstract double price();
    // [...]
}
```

Limousine.java

```
public class Limousine extends Car {
    public double price() {
        return 25000.00;
    }
}
```

Extra.java

```
public abstract class Extra extends Car {
}
```

AirCondition.java

```
public class AirCondition extends Extra {
    private Car car;
    public AirCondition(Car car) {
        this.car = car;
    }
    public double price() {
        return car.price() + 300.00;
    }
}
```