Package --- com.java.packagename Class A Class C [Zeug von eigener Klasse] [eigene Attribute und Methboden] #DrittesBeispiel [sichbar von Class A: +ViertesBeispiel ~ZweitesBeispiel #DrittesBeispiel +ÜberschreibeBeispiel() +ViertesBeispiel] +ÜberladeBeispiel() erbt von Legende: Class B (extends Class A) -private #protected [eigene Attribute und +public Methoden] +int eigenesAttribut public Athing = new B(); +Überladebeispiel(param) System. dut.println(thing.eigenesAttribut); +Überschreibebeispiel() public A thingzwei = new B(); [sichtbar von Klasse A: System.out. rintln((B)thingzwei.eigenesAttribut); ~ZweitesBeispiel → kann nich finden, gibt Fehler aus? (oder 0??) #DrittesBeispiel +ViertesBeispiel thingzwei.überschreibebeispiel(); +Überladebeispiel() +Überschreibebeispiel()] Package

Class D

[eigene Attribute und Methboden]

[sichbar von Class A: +ViertesBeispiel]

Class D extends A

[eigene Attribute und Methboden]

[sichbar von Class A:
#DrittesBeispiel
+ViertesBeispiel]

Code:

public Auto meinAuto = new Auto(); System.out.println(meinAuto.ps); meinAuto.beschleunigen();

Was passiert:

Instanzieren eines Auto-Objekts mit Namen "meinAuto" Ausgeben der PS von meinAuto. Aufruf der Methode beschleunigen von meinAuto.

Klasse:

Attribute: +int ps +String modell -float hubraum Auto() → Objekt! Auto(ps, modell, hubraum) Auto(ps) +beschleunigen() +bremsen() +hupen()

Objekt:

meinAuto (Objekt
der klasse AUTO)

ps = 200
modell =
"Alphawasauchimmer"
hubraum= 3.0

+beschleunigen()
+bremsen()
+hupen()

variable = 100;

System.out.println(variable ++)
100

→101

System.out.println(++variable) →102

→ 102 102

Speicher:

