



# Programme structure

Niklas Lohbusch, Laurence Meyer

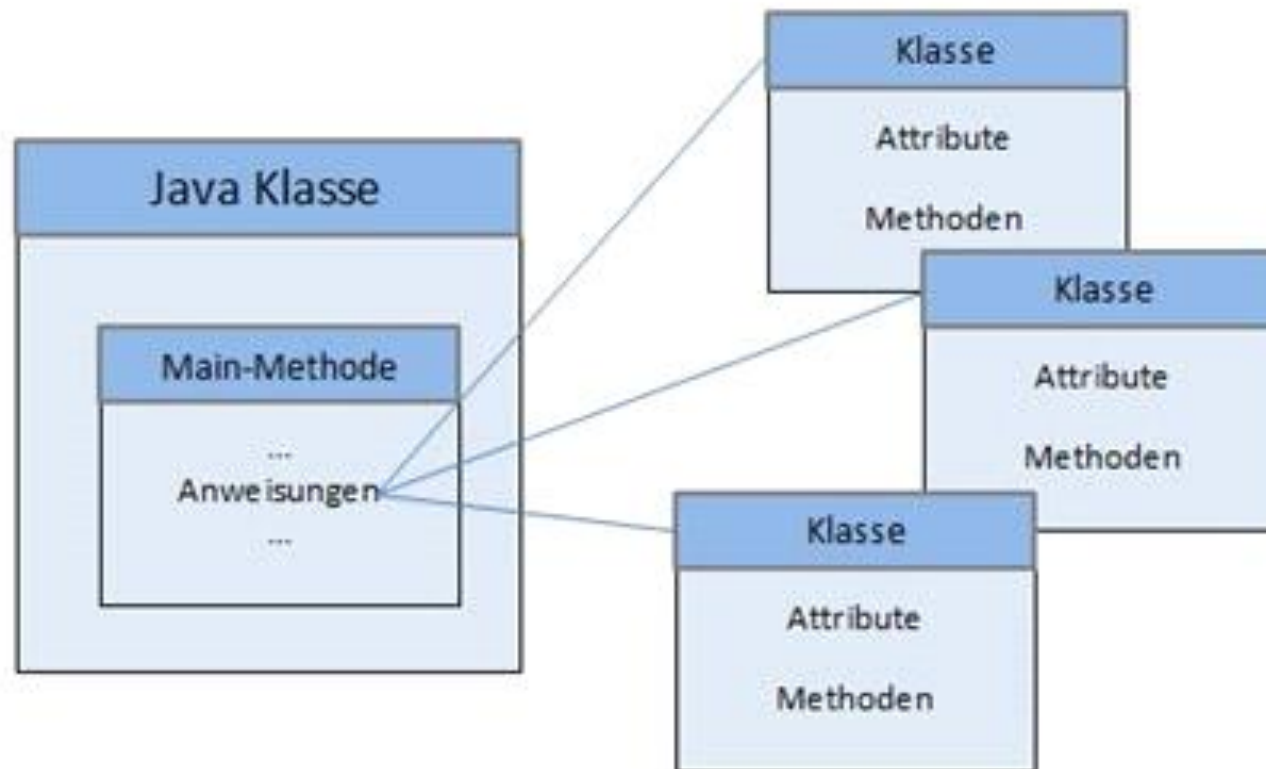
# Schedule \_

- 1. class
- 2. method
- 3. block nesting
- 4. default package
- 5. import
- 6. sources

# class \_

- In Java ist alles assoziiert mit Klassen und Objekten
- diesen sind Attribute und Methoden untergordnet
- Beispiel: Ein Auto ( Klasse ) hat ein Gewicht und eine Farbe ( Attribute ) bzw. kann es beschleunigen und bremsen ( Methoden )

class \_



## method \_

- ist ein Code Block der nur dann ausgeführt wird wenn aufgerufen
- wie eine function
- zum Wiederverwenden von Code Blöcken

# block nesting \_

- verschiedene Code Blöcke können ineinander geschrieben werden
- auch wenn funktional, recht unübersichtlich
- man beachte die Braces {}

# default package \_

- Klasse ohne Paket Angabe kommt in das unbenannte Paket oder auch **default package**
- gut für experimentellen Code oder zu Anfang eines Projektes
- schlecht als Struktur: benannte Pakete sind übersichtlicher und führen nicht zu Konflikten
- default package Klassen können untereinander interagieren
- benannte Pakete können nicht mit default package Klassen interagieren

# import \_

- Möglichkeit Klassen, Pakete oder Methoden zu importieren
- Zugriff aus anderen Klassen & Paketen möglich
- Beispiel: `java.util.*` , `Scanner`, `StringJoiner`, `Random`, ...



# Sources —

- <https://stackoverflow.com/questions/2335211/what-is-the-default-package-in-which-my-classes-are-put-if-i-dont-specify-it>
- [https://javabeginners.de/Klassen\\_und\\_Interfaces/Innere\\_Klassen.php](https://javabeginners.de/Klassen_und_Interfaces/Innere_Klassen.php)
- <https://www.youtube.com/watch?v=Y7sWSCIPFSw>
- [https://openbook.rheinwerk-verlag.de/javainsel/03\\_006.html#u3.6.7](https://openbook.rheinwerk-verlag.de/javainsel/03_006.html#u3.6.7)