



# Autohändler Beispiel

- Aufgabenstellung
- UML Diagramm
- Implementierungsdetails

# Aufgabenstellung



- Autos, Motorräder, Oldtimer
- **Auto:** Name, Erstzulassung, Leistung, Spritverbrauch, Anzahl der Sitze, EurotaxJEinkaufspreis, tatsächlicher Einkaufspreis, EurotaxJVerkaufspreis
- **Motorrad:** Name, Erstzulassung, Leistung, Spritverbrauch, EurotaxJEinkaufspreis, tatsächlicher Einkaufspreis, EurotaxJVerkaufspreis, Motortyp (2takt, 4takt)
- **Fortbewegungsmittel (Interface!):** Autos, Motorräder
  - Methode zur Berechnung der Passagierkapazität
  - Methode zur Berechnung des Kilometerpreises (nur Benzin)

# Aufgabenstellung



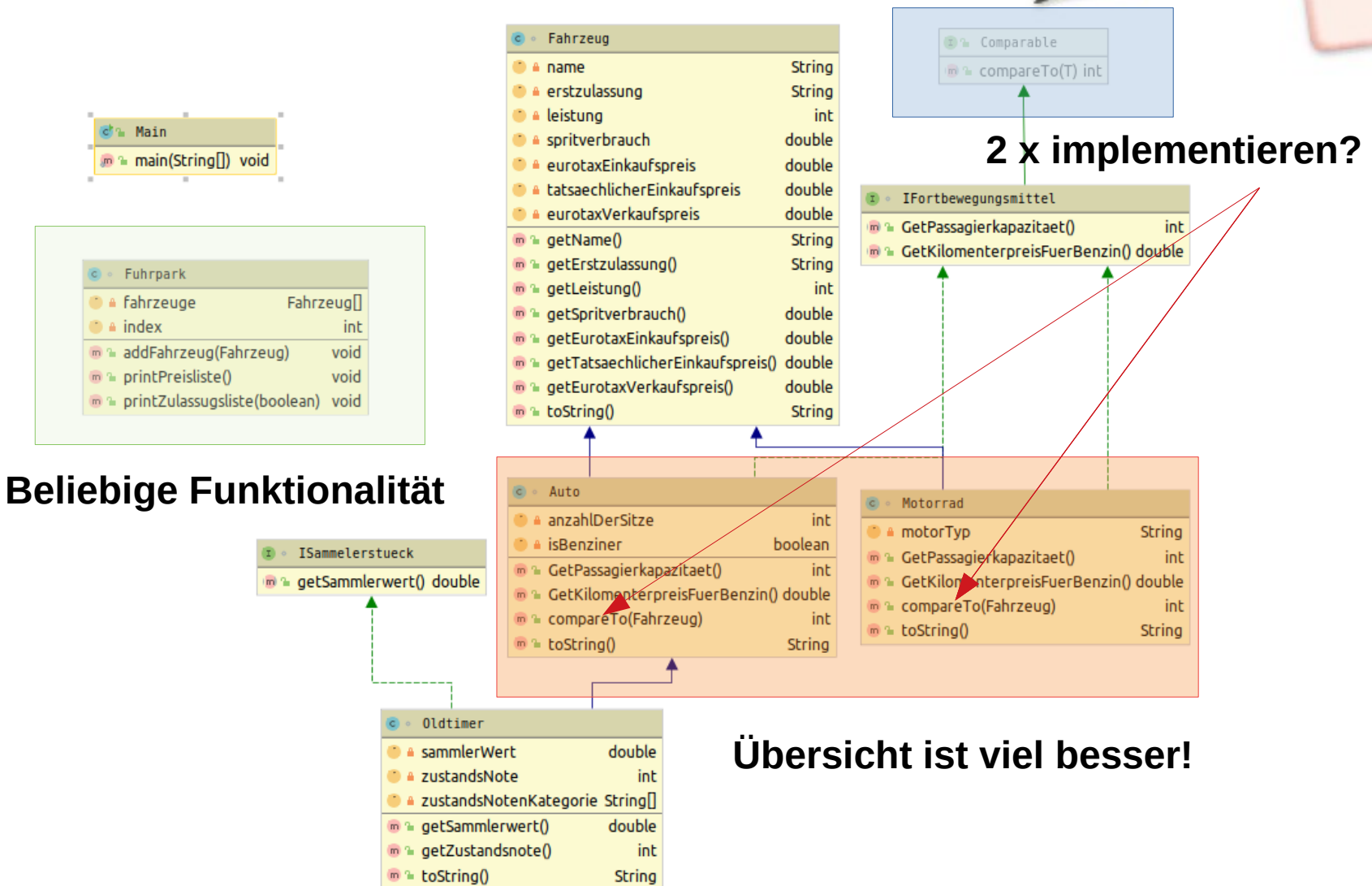
- Oldtimer **sind** spezielle Autos, welche Sammlerstücke (**Interface**) sind!
  - Methode zur Berechnung des Sammlerwert
- Oldtimer haben eine Zustandsnote
- Fuhrpark mit einem passenden Array.
- Fahrzeuge sollen als Liste sortierbar angezeigt werden.

# KISS





# UML Diagramm - Detail



# Implementierungsdetails



```
public class Main {
```

**Standard: Objekte erzeugen!**

```
public static void main(String[] args) {  
    Auto fiat127 = new Auto("Fiat 127", "1982-12-31", 34, 5.3, 6000, 6300, 7000, 5, true);  
    Auto manta = new Auto("Manta Gold", "1977-01-31", 66, 15.3, 16000, 16300, 17000, 5, true);  
    Motorrad yahmaha03 = new Motorrad("Yahama S1000", "2018-06-28", 40, 8.3, 4000, 4100, 5000, "2-Takt");  
    Oldtimer rolls = new Oldtimer("The Diamant", "1950-04-27", 66, 23.0, 28000, 41000, 55000, 5, true, 100000, 2)  
  
    Fuhrpark fuhrpark = new Fuhrpark(4);  
    fuhrpark.addFahrzeug(manta);  
    fuhrpark.addFahrzeug(fiat127);  
    fuhrpark.addFahrzeug(yahmaha03);  
    fuhrpark.addFahrzeug(rolls);  
    fuhrpark.addFahrzeug(manta);  
  
    fuhrpark.printPreisliste();  
    System.out.println();  
    fuhrpark.printZulassugsliste(true);  
    System.out.println();  
    fuhrpark.printZulassugsliste(false);  
}
```

**Liste sortieren mit Comparable**

**Listen sortieren mit Comparator**

# Comparable vs Comparator



```
import java.util.Comparator;

public class ZulassungsComparer implements Comparator<Fahrzeug> {

    @Override
    public int compare(Fahrzeug o1, Fahrzeug o2) {
        return o1.getErstzulassung().compareTo(o2.getErstzulassung());
    }
}
```

```
class Auto extends Fahrzeug implements IFortbewegungsmittel
{
    ....

    @Override
    public int compareTo(Fahrzeug o) {
        return (int) (this.getEurotaxVerkaufspreis() - o.getEurotaxEinkaufspreis());
    }

    ....
}
```