LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

mobile and distributed systems group

Javakurs für Anfänger

Einheit 03: Wiederholung Klassen & Objekte

Kyrill Schmid
Lehrstuhl für Mobile und Verteilte Systeme

Heutige Agenda

Wiederholung

- Klassen, Objekte, Attribute und Methoden
- Das Schlüsselwort `this`
- Programmieraufgabe: Kreisberechnung

Praxis:

- Wiederholung
- Die Klasse Kreis schreiben

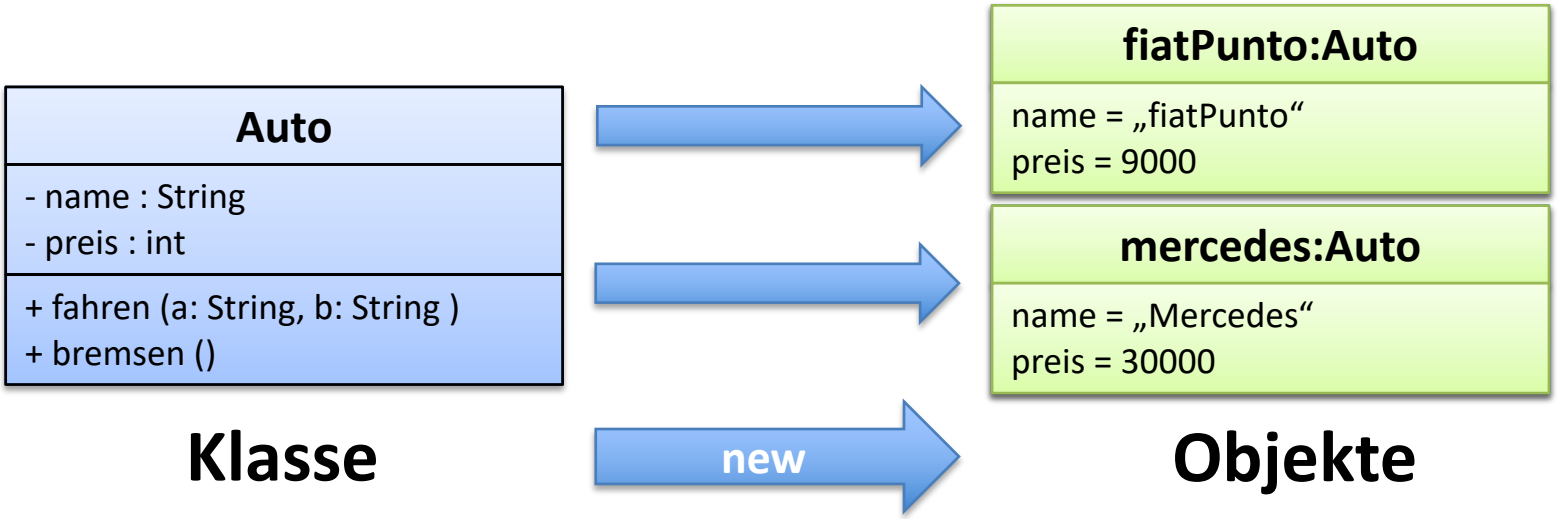
Wiederholung: Klassen & Objekte

Klasse:

- Stellt ein **Konzept** bzw. **Bauplan** dar
- Beschreibt dadurch einen Teil der Realität (**Attribute** und **Methoden**)

Objekt (= Instanz einer Klasse):

- Wird beim Ausführen des Programms (gemäß nach dem Bauplan der Klasse) erzeugt und spätestens beim Beenden wieder verworfen
- Bekommt **Werte** für seine Attribute



Wiederholung: Attribute & Methoden

Attribute werden durch **Instanzvariablen** definiert

- Direkten Zugriff von außerhalb der Klasse vermeiden!
  - Prinzip der Datenkapselung
  - Daher: `private` Deklaration
  - Beispiel:
    - `private String name;`
    - `private int preis;`

Das Verhalten wird durch **Methoden** definiert:

```
<Modifier> Rückgabetyp Methodenname(Parametertyp Parameter1,…){
      //Methodenrumpf
}
```

Verwendung und Manipulation eines Objekts über dessen Methodenaufrufe

Wiederholung: Konstruktoren

Wir benötigen einen **Konstruktor** zum Erzeugen von Objekten

- Aufgabe: Bringe das Objekt in einen definierbaren **Anfangszustand** beim Erzeugen
  - Anfangsbelegung der Instanzvariablen
  - Kann frei vom Entwickler programmiert werden
  - Ist eine spezielle Methode der Form:

```
public Klassenname (Parametertyp Parameter1,…){
  //Anweisungen
}
```

- Es können mehrere Konstruktoren definiert werden
  - Der „richtige“ Konstruktor wird nach Anzahl und Typen der Parameter aufgerufen
    - Dazu später mehr…
- Ist kein Konstruktor definiert wird vom Compiler automatisch ein Default-Konstruktor erzeugt
  - Besitzt keine weitere Anweisungen
- Der entspr. Konstruktor wird bei der Objekterzeugung mit `new` aufgerufen

Wiederholung: Das Schlüsselwort `this`

Das Schlüsselwort `this` wird als Selbstreferenz bezeichnet

- Verweist immer auf das eigene Objekt
- Um auf Instanzvariablen zuzugreifen: `this.instanzvariable`
  - Beugt versehentliche Verwechslungen mit lokalen Variablen oder Parametern vor
- Beispiel:

```
public class Person{
//Instanzvariablen
   private String meinName;
   private int alter;


   public Person(String name){
      meinName = name;
   }
}
```

Instanzvaribale **meinName** wird durch Parameter **name** nicht überschattet! Daher: kein `this` nötig aber möglich!

Richtig wäre: `this.meinName = meinName;`

```
public class Person{
//Instanzvariablen
   private String meinName;
   private int alter;


   public Person(String meinName){
      meinName = meinName;
   }
}
```

Instanzvaribale **meinName** wird durch Parameter **meinName** überschattet! Daher: *this* unbedingt nötig, um Instanzvariable zu belegen!

Eigene Aufgabe

Versuchen Sie nun selbstständig die folgende Programmieraufgabe:

Schreiben Sie eine Klasse ***Student*** mit den folgenden Attributen und Methoden:

- Attribute:
  - Name (`String`)
  - Semester (`int`)
- Methoden:
  - `public int welchesSemester()`
    - Gibt das aktuelle Semester des Studenten zurück
  - `public void lernen()`
    - Soll auf der Konsole ausgeben: „… lernt gerade“.

Schreiben Sie innerhalb der Klasse eine Main-Methode und erzeugen Sie darin 2 Studenten mit jeweils einem Namen und der Semesterzahl.

Lassen Sie sich für beide Studenten die Semesterzahl auf der Konsole ausgeben.

Rufen Sie für beide Studenten die Methode lernen() auf.