OOP

Vererbung

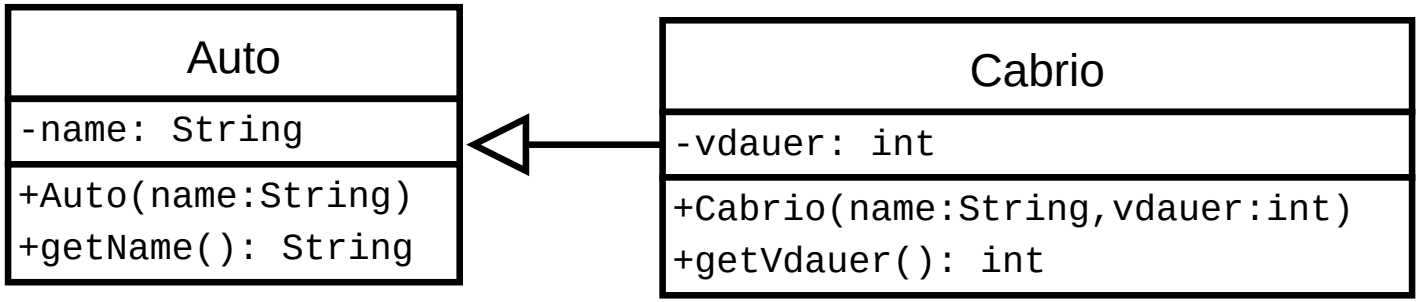
Mathias Hanacek

hanacek@elektronikschule.de

Elektronikschule Tettnang

Wiederverwendung

- Verwendung einer vorhandenen Superklasse als ’Blaupause’ für eine neue Subklasse

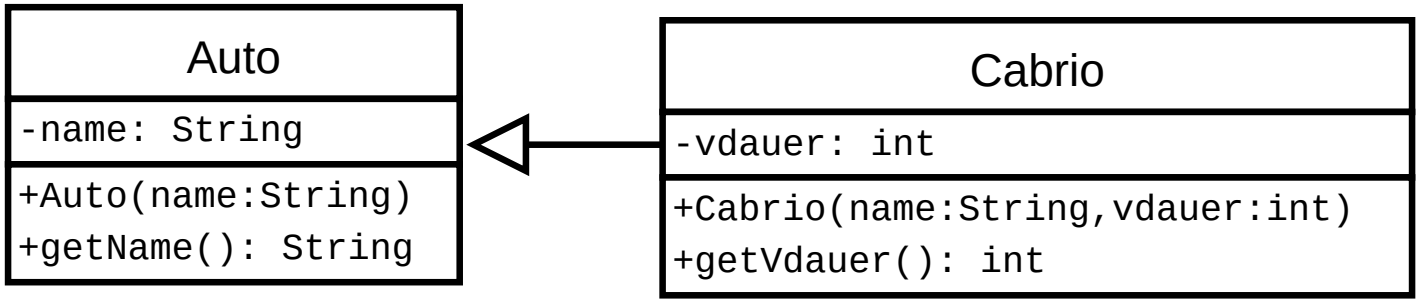


Superklasse

Subklasse

Wiederverwendung

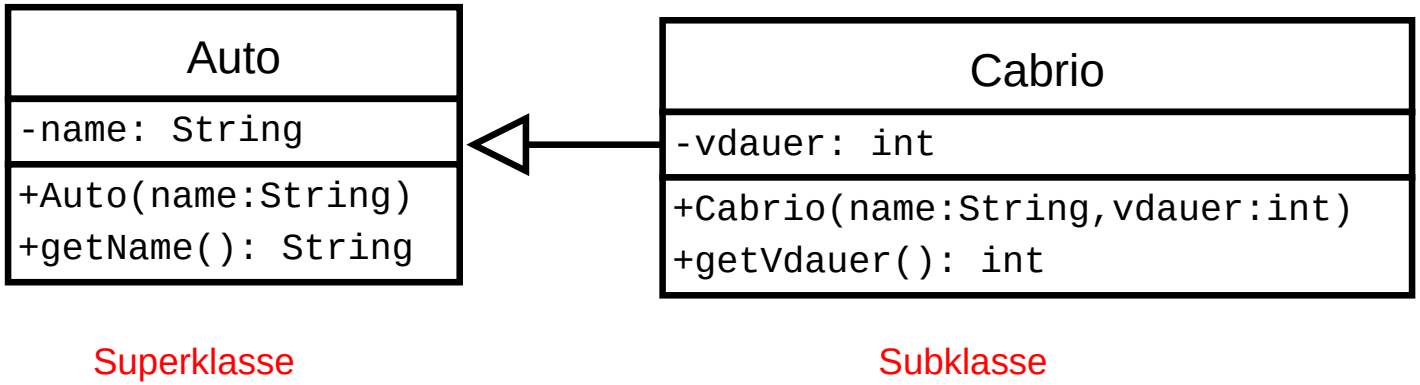
- Verwendung einer vorhandenen Superklasse als ’Blaupause’ für eine neue Subklasse



- Vererbt werden: Attribute und Methoden

Wiederverwendung

- Verwendung einer vorhandenen Superklasse als ’Blaupause’ für eine neue Subklasse



- Vererbt werden: Attribute und Methoden
- **Nicht** vererbt werden: Konstruktoren

im Code

```
public class Auto {
  private String name;
  public Auto(String name) {
    this.name = name;
  }
  public String getName() {
    return name;
  }
}
```

```
public class Cabrio extends Auto {
  private int vdauer;
  public Cabrio(String name,
                int vdauer) {
    //FALSCH! this.name = name;
    super(name);
    this.vdauer = vdauer;
  }
  public int getVdauer() {
    return vdauer;
  }
}
```

Wie wird Vererbung verwendet?

```
Auto ente = new Cabrio("Ente", 30);
Auto aklasse = new Auto("Aklasse");
Cabrio kaefer = new Cabrio("Kaefer", 30);
Cabrio smart = new Cabrio("Smart");
Cabrio kadett = new Auto("Kadett");

int d1 = kaefer.getVdauer();
int d2 = ente.getVdauer();
String n1 = ente.getName();
String n2 = kaefer.getName();

boolean b1 = ente instanceof Cabrio;
boolean b2 = aklasse instanceof Cabrio;
boolean b3 = ente instanceof Object;
```

korrekte Verwendung

korrekt

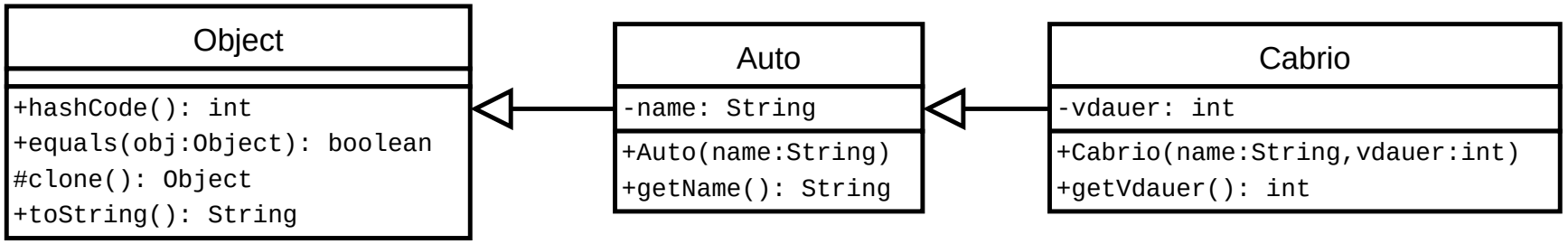
```
Auto ente = new Cabrio("Ente", 30);
Auto aklasse = new Auto("Aklasse");
Cabrio kaefer = new Cabrio("Kaefer", 30);


int d1 = kaefer.getVdauer();
String n1 = ente.getName();
String n2 = kaefer.getName();


//true:
boolean b1 = ente instanceof Cabrio;
boolean b3 = ente instanceof Object;
```

falsch

```
Cabrio smart = new Cabrio("Smart");
Cabrio kadett = new Auto("Kadett");



int d2 = ente.getVdauer();



//false:
boolean b2 = aklasse instanceof Cabrio;
```

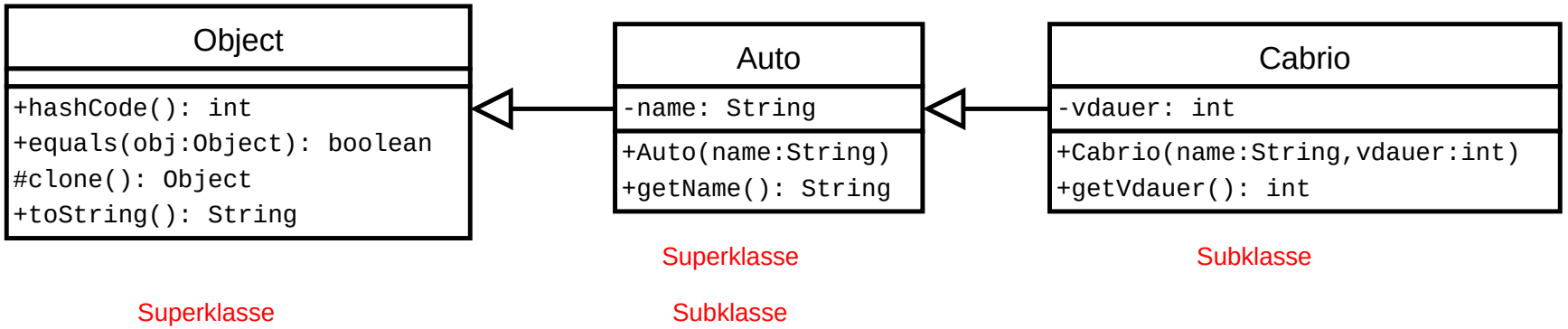
Die Superklasse Object

- Ohne Angabe von extends wird automatisch Object als Superklasse verwendet



Die Superklasse Object

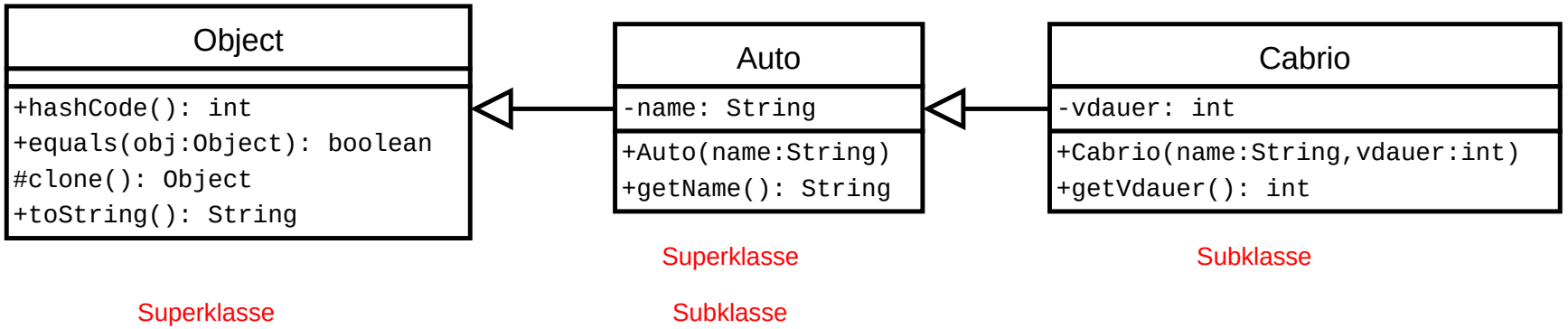
- Ohne Angabe von extends wird automatisch Object als Superklasse verwendet



- Ausgabe? `System.out.println(ente);`

Die Superklasse Object

- Ohne Angabe von extends wird automatisch Object als Superklasse verwendet



- Ausgabe? `System.out.println(ente);`
- Überschreiben Sie in der Klasse **Auto** die geerbte Methode **toString** und testen Sie erneut.

Aufgabe Geometrie

Es sollen 5 Klassen für folgende geometrische Objektarten erstellt werden:
Parallelogramm, Rechteck, Quader, Quadrat, Würfel.
Alle zweidimensionalen Objektarten sollen außer dem Konstruktor nur zwei parameterlose Methoden zur Berechnung und Rückgabe

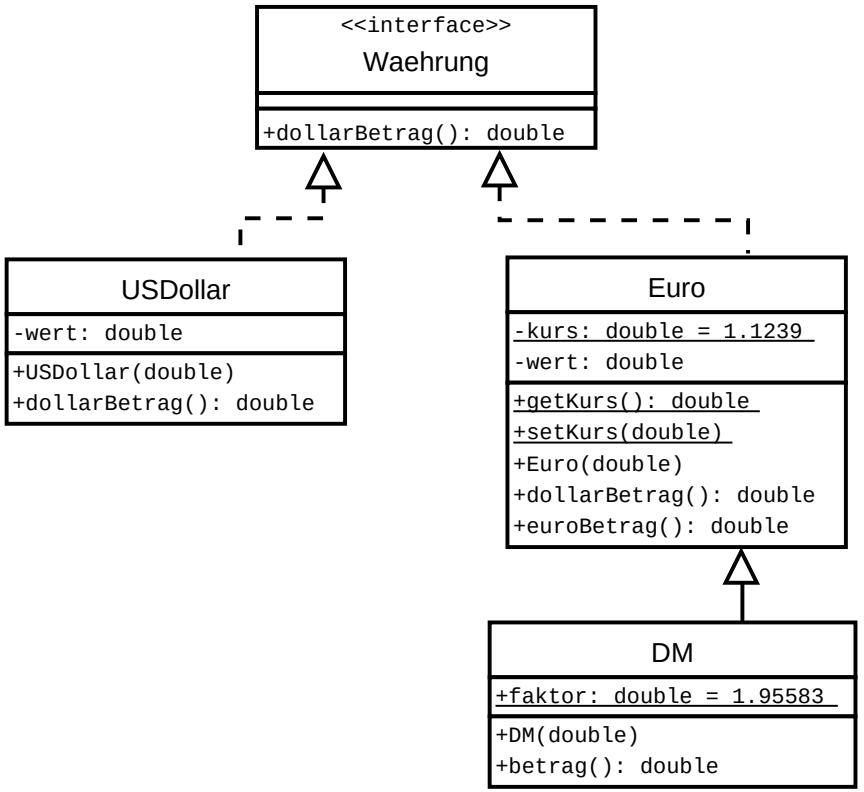
- des Flächeninhalts (Methodenname flaeche)
- des Umfangs (Methodenname umfang)

besitzen.
Die dreidimensionalen Objektarten sollen neben dem Konstruktor nur eine Methode volumen zur Berechnung und Rückgabe des Volumens besitzen.
Die Konstruktoren sollen so wenig wie möglich Übergabeparameter haben.
Es sollen nur Werte für Eigenschaften übergeben werden, die zur Berechnung in den Methoden flaeche, umfang bzw. volumen notwendig sind. Beim Parallelogramm wären das z.B. die Seitenlänge a, die Seitenlänge b und die zur Seite a senkrechte Höhe ha.

Zeichnen Sie das UML-Klassendiagramm und setzen Sie den Entwurf in Code um.

Aufgabe Währung

Implementieren Sie den folgenden Entwurf:



Zum Test wird Ihnen eine Klasse *Buchhaltung* zur Verfügung gestellt.