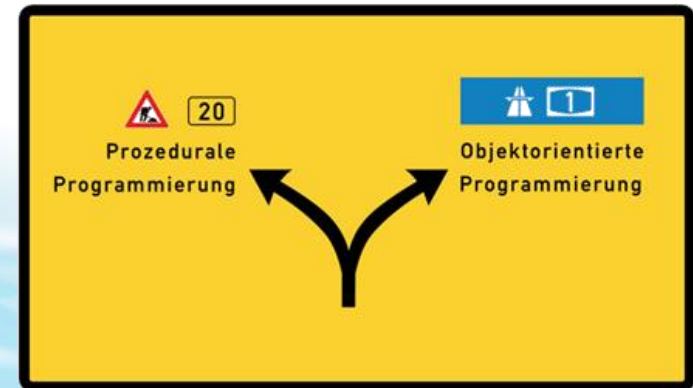


Java-Grundlagentraining

Tag 2

SHD Einzelhandelssoftware
GmbH & Co. KG

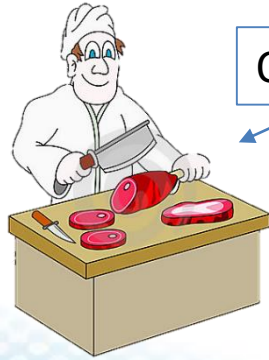
SHD



Agenda

- Die Objektorientierung
- Was ist Objektorientierte Programmierung
- Erstellen einer Struktur in IntelliJ
- Klassendiagramm
- Aufgabe 1 - Erstellen von Klassen
- Objekte erzeugen / instanziiieren
- Aufgabe 2 - Arbeiten mit Objekten
- Service / Controller Klassen
- Automaten Struktur anlegen
- Aufgabe 3 - Programmierung des FutterAutomat
- Hausaufgabe

Die Objektorientierung

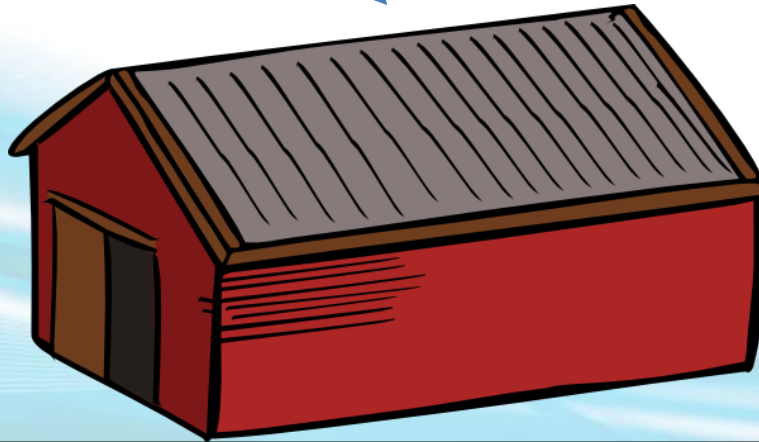


Objekt

Objekt



Objekt



Objekt



Was ist Objektorientierte Programmierung

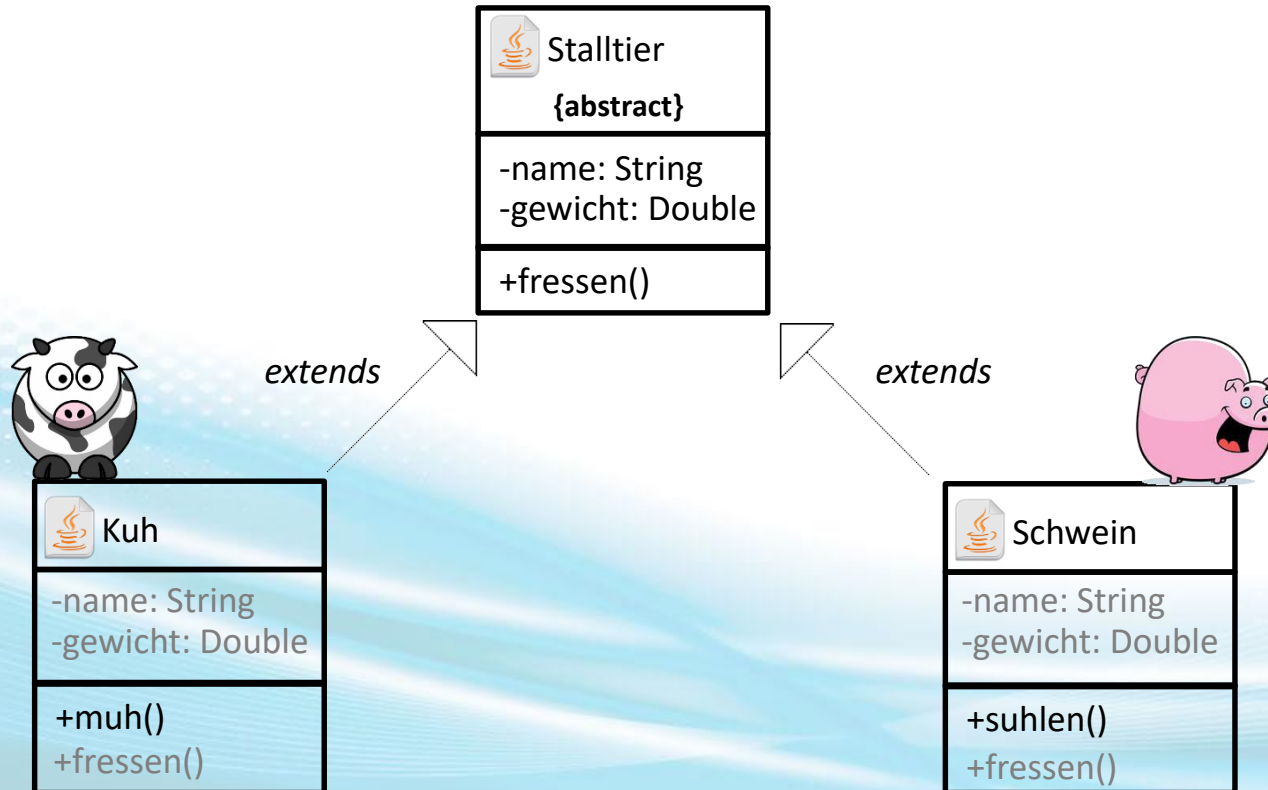
Jede objektorientierte Programmiersprache biete folgende Merkmale:

1. **Zusammenfassung von Daten (Attribute) und Funktionen (Methoden)**
In objektorientierten Sprachen werden Attribute und Methoden in „*Klassen*“ zusammengefasst. Von Klassen werden „*Instanzen*“ erzeugt, um die Attribute über die Methoden zu bearbeiten.
2. **Kapselung**
Die Attribute innerhalb einer Klasse sollen nur über ihre Methoden geändert werden können.
3. **Vererbung**
„*Wie der Vater, so der Sohn*“
Eine abgeleitete Klasse erbt die Methoden und Attribute der Basisklasse
4. **Polymorphie (Vielgestaltigkeit)**
„*Verschiedene Stalltiere werden vom selben Metzger geschlachtet*“
Die Möglichkeit je nach Verwendung des Objekts genau einen bestimmten Datentyp annehmen

Erstellen einer Struktur in IntelliJ



Klassendiagramm



Aufgabe 1 - Erstellen von Klassen

Folgende Aufgaben sind zu erledigen:

1. Erstelle einen Konstruktor für Kuh und Schwein in dem das Gewicht unterschiedlich gesetzt wird.
2. Erzeuge die Getter und Setter für die Klassenattribute
3. Implementiere ein Defaultverhalten für die Methode fressen() im Stalltier und eine explizites Verhalten für die Methode fressen() im Schwein.
Das Schwein soll die doppelte Gewichtszunahme haben.
4. Erstelle die Methoden muh() & suhlen().
Es soll der Name des Tieres sowie eine entsprechende Nachricht ausgegeben werden.

Objekte erzeugen / instanziiieren

```
Kuh kuh = new Kuh();
```

```
Schwein schwein = new Schwein();
```

//Polymorphie

```
Stalltier kuh2 = new Kuh();
```

```
Stalltier schwein2 = new Schwein();
```

```
Stalltierstalltier = new Stalltier(); // weil abstract
```

```
kuh.fressen();
```

```
schwein.fressen();
```

```
kuh.muh();
```

```
schwein.suhlen();
```


Benutzen eine Liste in Java (Basics)

```
List<Stalltier> stalltierList = new ArrayList();  
  
//Element hinzufügen:  
stalltierList.add(new Kuh());  
  
//Element löschen (mit Objekt oder Index):  
stalltierList.remove(kuh) oder stalltierList.remove(0);  
  
//Element auslesen mit dem Index:  
stalltierList.get(0);  
  
//Anzahl der Elemente einer Liste ermitteln:  
stalltierList.size();
```

Aufgabe 2 - Arbeiten mit Objekten

Erledige folgende Aufgaben:

1. Befülle eine Liste zufällig (ob Schwein oder Kuh) mit 10 Stalltieren. (z.b. mit `Math.random()`)
2. Iteriere (for-Schleife) über die erstellte Liste von Stalltieren und gebe dabei ihren Namen aus.
3. Iteriere (for-each-Schleife) über die erstellte Liste von Stalltieren und lasse sie dabei fressen.
4. Wenn das Stalltier fertig mit fressen ist, rufe je nach Typ entweder „muh“ oder „suhlen“ auf.
(z.b. mit `instance of`)

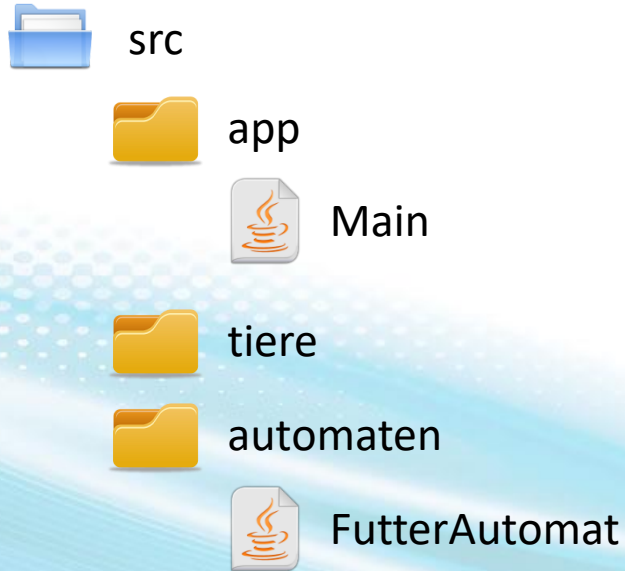
Service / Controller Klassen

1 Anwendungsfall = 1 Service

Anwendungsfall „Füttern“



Automaten Struktur anlegen



Aufgabe 3 - Programmierung des FutterAutomat

Aufgabe:

1. Lagere die Fütterlogik in den FutterAutomat aus.
2. Erstelle in der main() Methode eine Instanz des FutterAutomaten und rufe den Füttermechanismus mit einer Liste von 10 zufälligen Stalltieren auf.

Hausaufgabe

Alle gelernten Inhalte in einem Projekt

Aufgabe:

- Erweitere den Stall um zwei zusätzliche Tiere: Schaf, Hund
- Das Schaf soll in den Fütterprozess mit eingebunden werden
- Der Hund soll nach dem Fütterprozess jedes Tier beißen
- Das bewirkt, dass sich das Tier bewegen muss, dies hat zur Folge, dass es etwas Gewicht verliert.