

The GAME Engineers

#6 – OOP und Klassen



The forging engineers



Inhalt

- OOP
- Klassen
 - Member
 - Methoden
 - Objekte
 - Konstruktoren

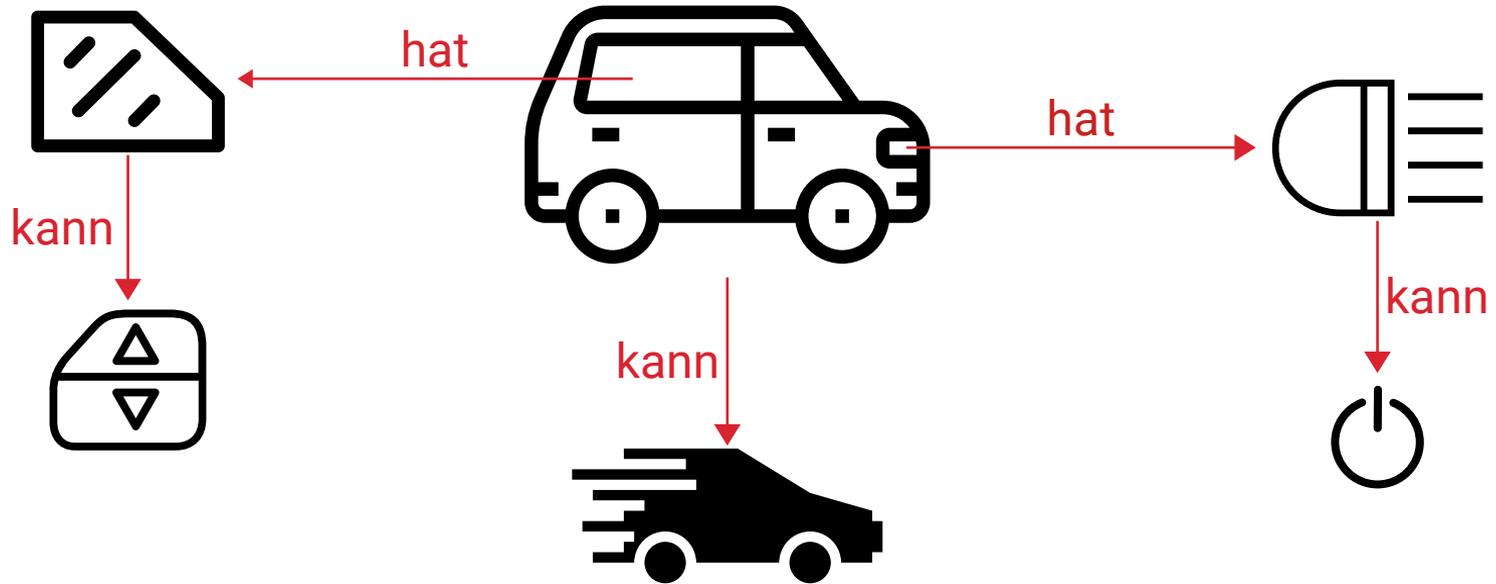


The forging engineers



OOP

- Man denkt in Objekten,
- die nach der Wirklichkeit modelliert sind

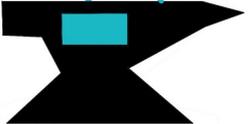


Klassen und Objekte

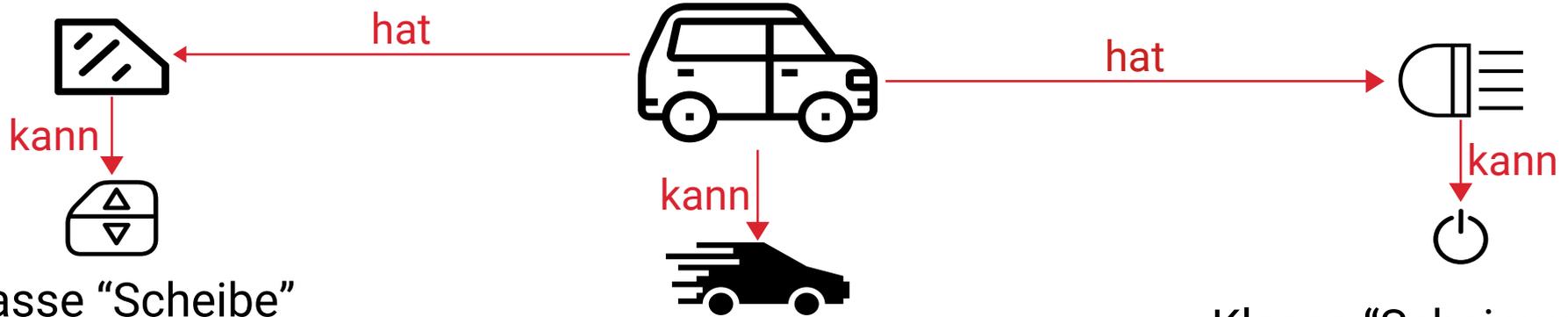
- Klassen sind “Baupläne” für Objekte
- Klassen beinhalten
 - “Members” (Variablen) und
 - Methoden



The forging engineers



Klassen und Objekte



Klasse "Scheibe"

- Member:
 - aktuellGeoeffnet
- Methoden:
 - auf
 - zu

Klasse "Auto"

- Member:
 - Scheibe
 - Scheinwerfer
- Methoden:
 - fahren

Klasse "Scheinwerfer"

- Member:
 - Kaputt
 - Licht
- Methoden:
 - an
 - aus



The forging engineers



Klassen und Objekte

- Member sind "Eigenschaften"
 - Was hat das Objekt?
- Methoden bestimmen das Verhalten
 - Was kann das Objekt?



The forging engineers



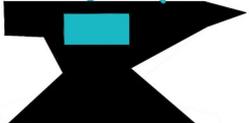
Klassen und Objekte

- Es können mehrere Objekte *instanziiert* werden
 - BMW baut von einem Modell (**Klasse**) ja auch viele Autos (**Objekte**)
 - Aber z.B. in versch. Farben, versch. Motorisierung, etc. (**Eigenschaften**)
 - Verhalten tun sie sich allerdings (erstmal) gleich, z.B. fahren (**Methoden**)

→ Objekte haben also je einen **eigenen** Zustand (**State**) und können eindeutig identifiziert und „angesprochen“ werden (**Referenz**)



The forging engineers



Klassen – Klassen

```
public class Auto {  
    public Scheibe scheibe = new Scheibe();  
    public Scheinwerfer licht = new Scheinwerfer();  
  
    public void fahren() {  
        System.out.println("Brumm Brumm");  
    }  
}
```

```
public class Scheibe {  
    public int aktuellGeoeffnet = 0;  
  
    public void auf(int wieWeit) {  
        aktuellGeoeffnet += wieWeit;  
    }  
  
    public void zu(int wieWeit) {  
        aktuellGeoeffnet -= wieWeit;  
    }  
}
```

```
public class Scheinwerfer {  
    public boolean istKaputt = false;  
    public boolean licht = false;  
  
    public void an() {  
        licht = true;  
    }  
  
    public void aus() {  
        licht = false;  
    }  
}
```



The forging engineers

→ Beachte, dass *static* nirgends steht!
Warum? Ein anderes Mal mehr dazu ...

Klassen – Objekte

- Anlegen eines Objektes:
 - Datentyp: Klassenname (hier **Auto**)
 - Variablenname
 - **new** Klassenname()
- **new** gibt uns neuen Speicherplatz auf dem **Heap**
 - Der **Heap** ist ein Speicherbereich, in dem Objekte “leben”
 - Durch **new** wird dort Speicher “vergeben” und falls ein Objekt nicht mehr gebraucht wird, wird der Speicher automatisch wieder freigegeben (**Garbage Collection**)
 - Speicher ist für andere Objekte wieder verfügbar

```
public static void main(String[] args)
{
    Auto auto1 = new Auto();
    Auto auto2 = new Auto();
}
```



The forging engineers



Klassen – Konstruktoren

- Du möchtest z.B. beim Auto die Farbe oder Sitzplatzanzahl festlegen?
 - Dann nutze doch einfach **Konstruktoren** du Idiot!
- **Konstruktoren** sind besondere Methoden
 - Müssen gleich wie Klasse heißen und
 - Haben keinen Rückgabewert
 - Können beliebig viele Parameter haben (meist zum Initialisieren von Mitgliedern)

```
public class Auto
{
    public int sitzplaetze;
    public String farbe;

    public Auto()
    {
        this.sitzplaetze = 5;
        this.farbe = "schwarz";
    }

    public Auto(int sitzplaetze, String farbe)
    {
        this.sitzplaetze = sitzplaetze;
        this.farbe = farbe;
    }
}
```



The forging engineers



Klassen – Konstruktoren

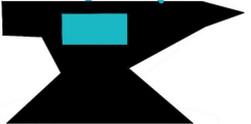
```
public static void main(String[] args)
{
    Auto auto1 = new Auto();
    Auto auto2 = new Auto(9, "gelb");
    Auto auto3 = new Auto(1, "weiss");

    System.out.println(auto1.sitzplaetze + ", " + auto1.farbe);
    System.out.println(auto2.sitzplaetze + ", " + auto2.farbe);
    System.out.println(auto3.sitzplaetze + ", " + auto3.farbe);
}
```

```
Ausgabe:
5, schwarz
9, gelb
1, weiss
```



The forging engineers



Klassen – bekannte Klassen

```
public static void main(String[] args)
{
    Scanner scan = new Scanner(System.in);
    int eingabe = scan.nextInt();

    Random rand = new Random();
    int zufall = rand.nextInt(5);

    String text = "Hallo"; // Spezialfall, ohne new

    int[] arr = new int[10]; // Arrays sind Objekte ohne Klasse
}
```



The forging engineers

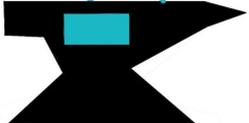


Klassen – Aufgabe

- Überlege bei dem Autobeispiel dir noch tiefergehende Member und Methoden.
→ Also modelliere das Objekt Auto weiter aus
gerne auch nebenher Coden, damit du die Syntax lernst :)



The forging engineers



Ende
Fragen?



The forging engineers

