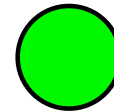
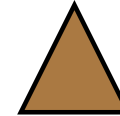


# Klassen



# Objekte



# Beispielprogramm: Exponat “Wilde Tiere!”



Was es tun soll:  
Sie berühren ein Bild.  
Das Tier brüllt und  
erzählt Ihnen etwas!

# Prozedural...



```
brüllen (String tierart) {  
    if (tierart.equals("Affe")) {  
        // Affensound laden und abspielen  
    } else if (tierart.equals("Löwe")) {  
        // Löwensound laden und abspielen  
    } else if (tierart.equals("Bär")) {  
        // Bärensound laden und abspielen  
    } ...  
}
```

# Objektorientiert...



Sagen Sie dem Affen direkt, was er machen soll!



# Objektorientiert...



Sagen Sie dem Affen direkt, was er machen soll!



Affe, brüll!

Affe, erzähl was!

“\$!@#%\$!!!#”

### **Klasse Affe**

```
brüllen () {  
    // Affensound laden und abspielen  
}  
  
erzählWas () {  
    // Informationen ausgeben  
}
```

### **Klasse Löwe**

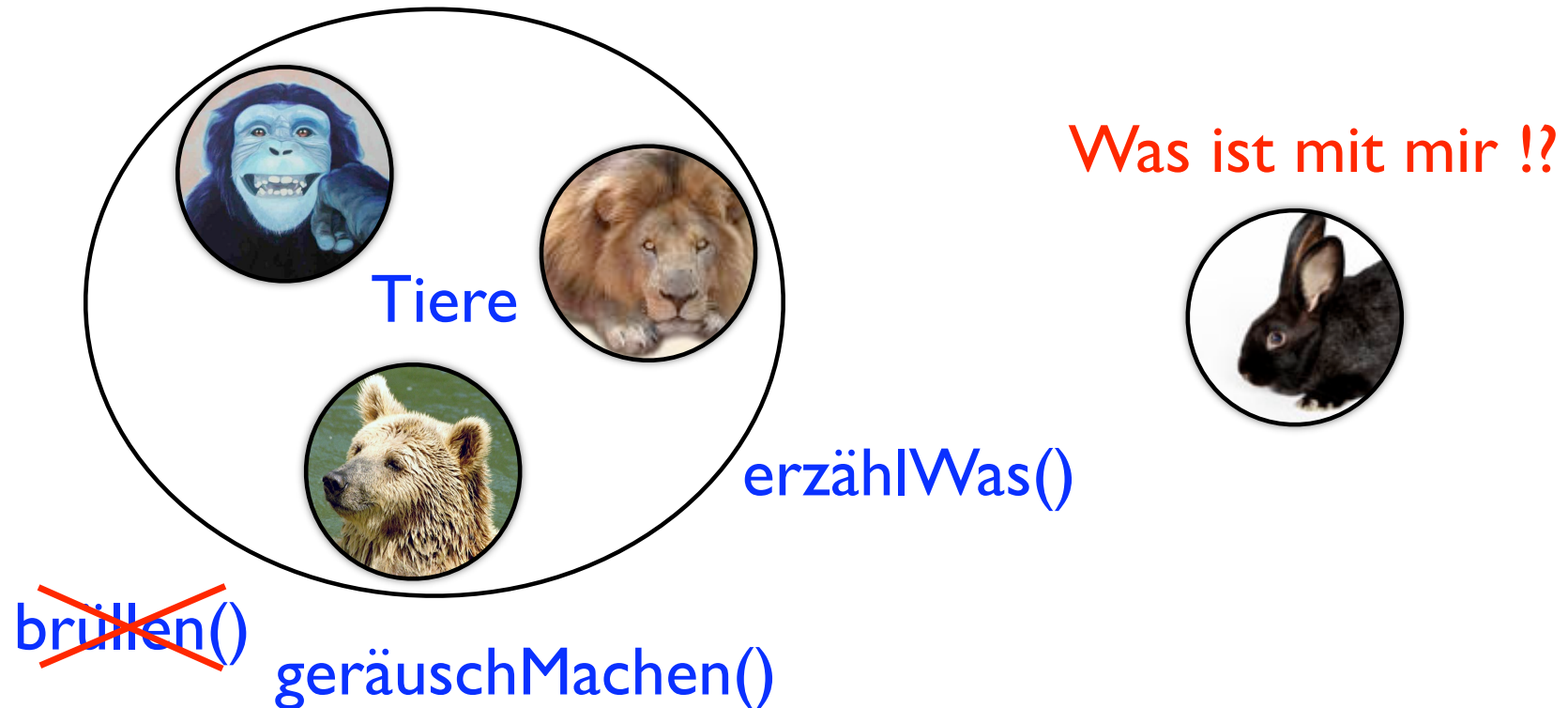
```
brüllen () {  
    // Löwensound laden und abspielen  
}  
  
erzählWas () {  
    // Informationen ausgeben  
}
```

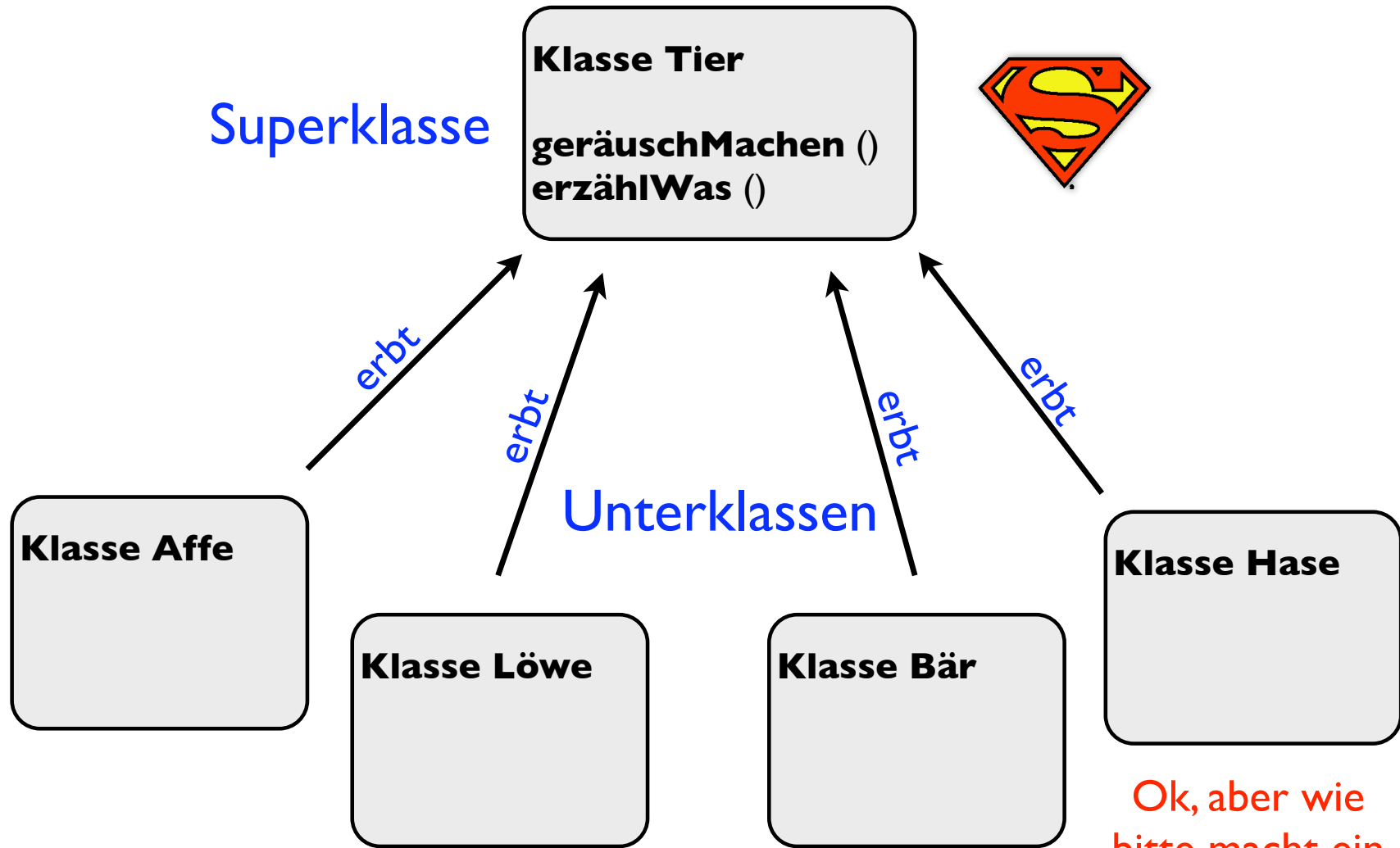
### **Klasse Bär**

```
brüllen () {  
    // Bärensound laden und abspielen  
}  
  
erzählWas () {  
    // Informationen ausgeben  
}
```

**Jedes Tier bekommt  
seine eigene Klasse!**

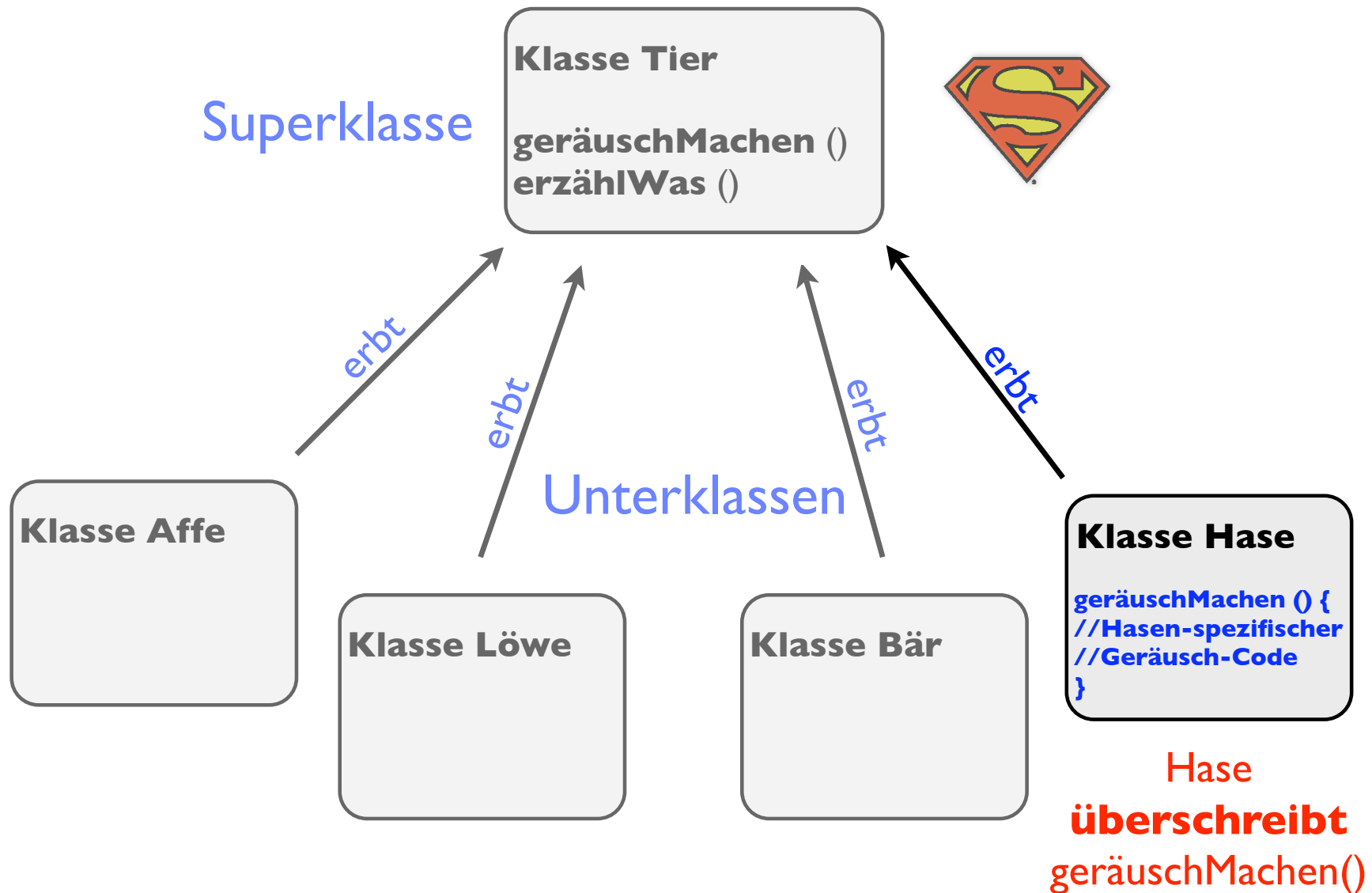
# Was haben Affen, Löwen und Bären gemeinsam?





Ok, aber wie  
bitte macht ein  
Hase Geräusche?





# Eine **Klasse** ist...



...wie die  
Zeichenschablone  
beim Malen...



...oder wie die  
Ausstechform beim  
Plätzchenbacken!

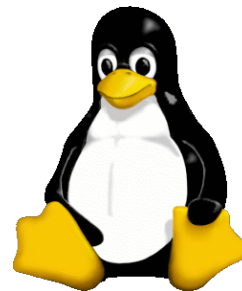
# Ein **Objekt** (eine Instanz) ist...



... ein bestimmtes  
Plätzchen...



...oder ein  
bestimmtes  
Auto...



...oder ein  
bestimmter  
Pinguin!

# Der Weg zur **Klasse**...

Think  
about it

...führt über ihre **Objekte**.

## **Instanzvariablen:**

das **Wissen** eines Bären

farbe

größe

hungrigSein

lieblingsEssen

imZirkusAufgetreten

## **Methoden:**

das **Verhalten** eines Bären

honigSuchen()

haseJagen()

tanzen ()

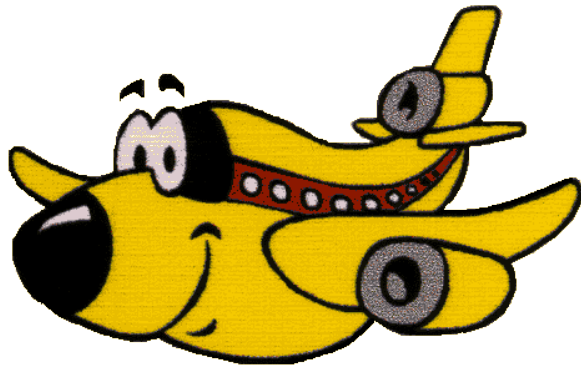
essen()

brüllen()



# Instanzvariablen ?

anzahlRäder    anzahlSitze  
typ    reichweite    anzahlMotoren  
abrißGeschwindigkeit  
...



# Methoden ?

landen ()    beschleunigen()  
abheben ()  
volltanken()    bremsen()  
autopilotFliegen ()    ...

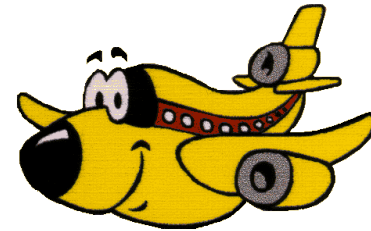
## Objekte mit **new** anlegen

```
Flugzeug a380 = new Flugzeug ();
```

```
Flugzeug b787 = new Flugzeug ();
```

↑  
Klasse (Typ)

↙  
Name des Objekts (Variable)



## Der **Punktoperator** .

```
a380.reichweite = 15000;
```

```
b787.reichweite = 15700;
```

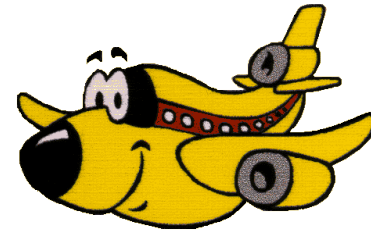
```
b787.abheben ();
```

greift auf **Instanzvariablen** und **Methoden** der Objekte zu.



# Objekte

## machen und testen



```
class Flugzeug {  
    // Instanzvariablen  
    int anzahlMotoren;  
    int reichweite;  
    String typ;  
  
    // Methoden  
    void abheben () {  
        System.out.println  
            ("Brrr.....zisch!!!!");  
    }  
    ...  
}
```

```
class FlugzeugTest {  
  
    public static void main (String[] args) {  
  
        // ein Flugzeug-Objekt erzeugen  
        Flugzeug a380 = new Flugzeug ();  
  
        a380.anzahlMotoren = 4;  
        a380.reichweite = 15000;  
        a380.typ = "Airbus A380";  
  
        a380.abheben ();  
    }  
}
```

# Was für Klassen/ Objekte brauchen wir?



Was es tun soll:  
Sie berühren ein Bild.  
Das Tier brüllt und  
erzählt Ihnen etwas!



## **class WildeTiereStarter**

```
main (String[] args) {  
    // Ein Zoo-Objekt erstellen  
    // Die Methode Zoo.losGehts() aufrufen  
}
```

## **class BenutzerEingabe**

```
// Berührtes Tier feststellen  
// und zurückliefern
```

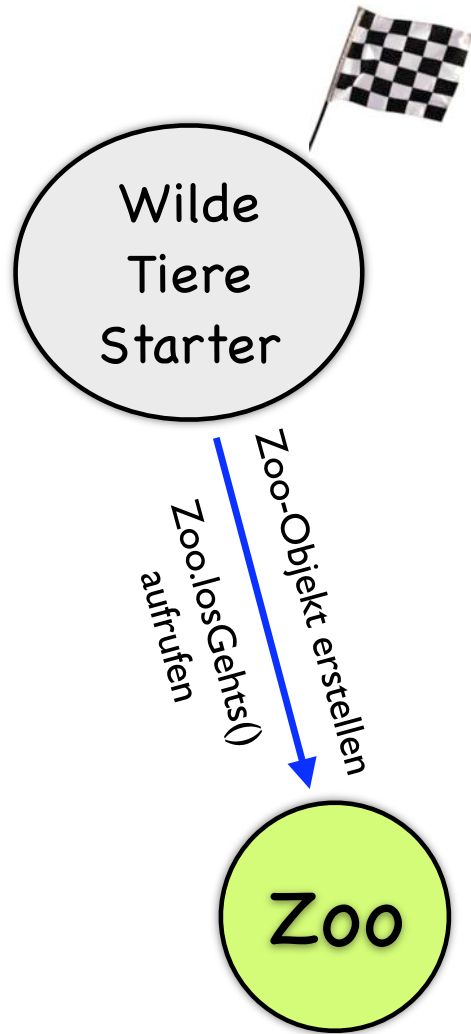
## **class Zoo**

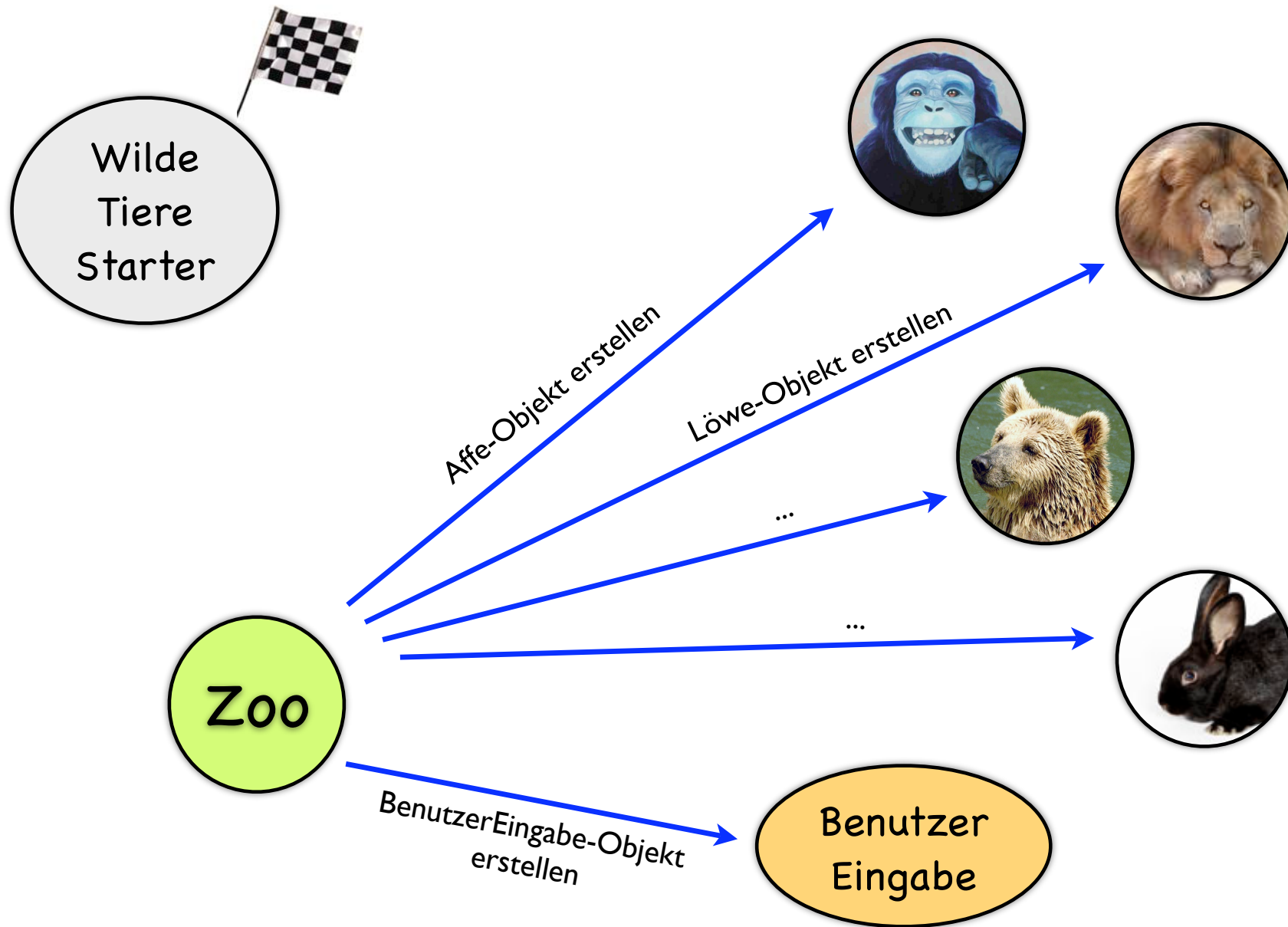
```
losGehts () {  
    // Tier-Objekte anlegen (also Affe, Löwe, Bär, Hase ...)  
    // BenutzerEingabe-Objekt erstellen  
  
    // Liefert die BenutzerEingabe ein Tier zurück,  
    // dann diesem Tier sagen, daß es Geräusche machen  
    // und etwas erzählen soll.  
}
```

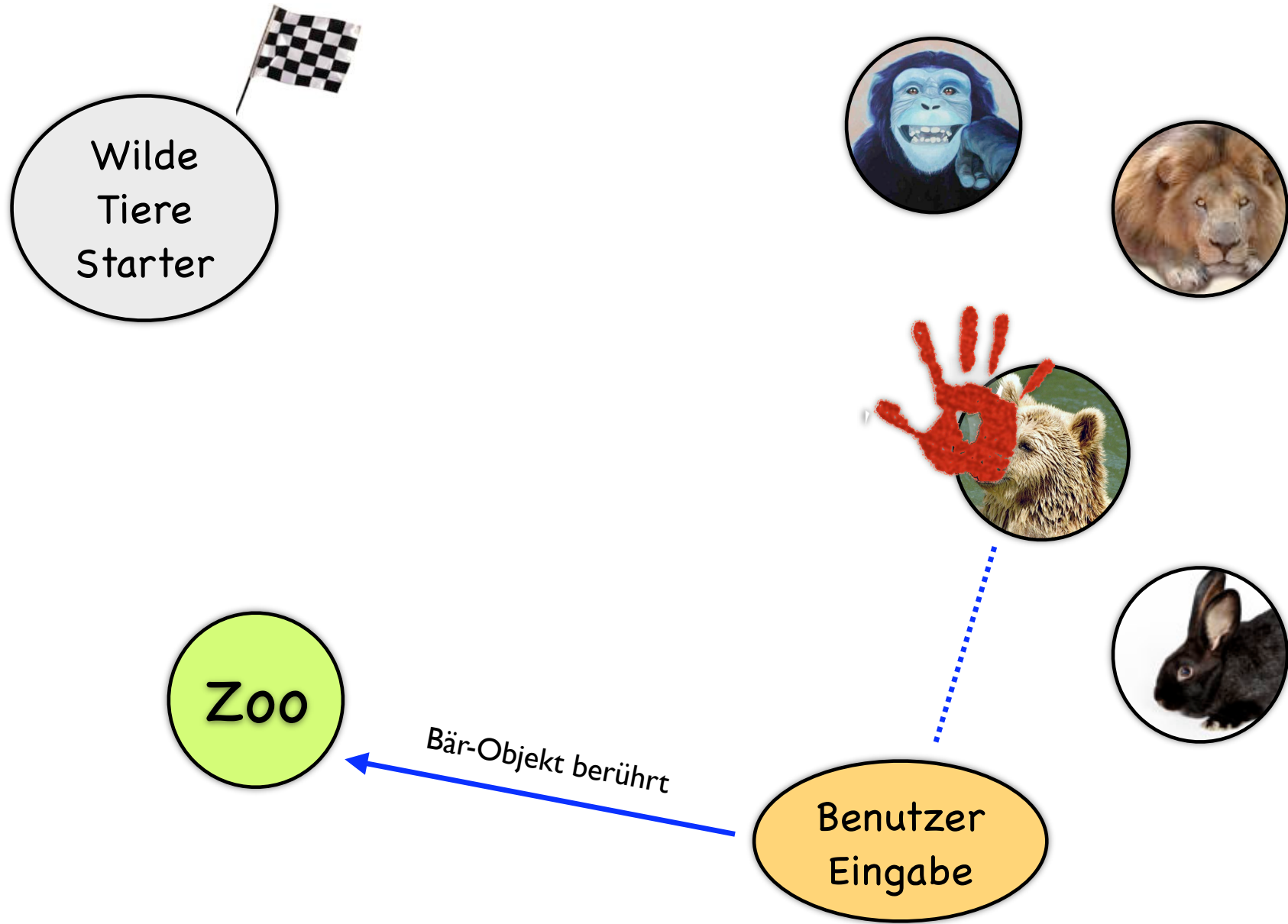
## **class Tier ...**

## **class Affe ...**

## **class Löwe ...**





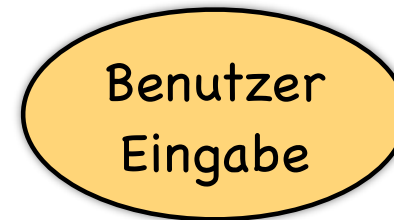
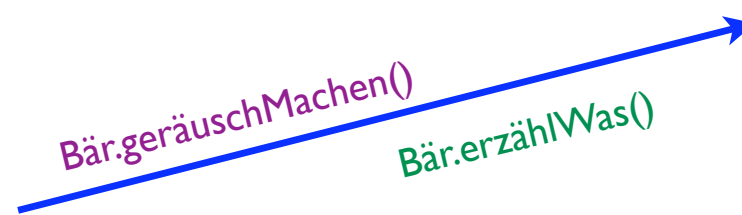
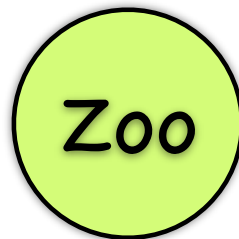




Grööö!



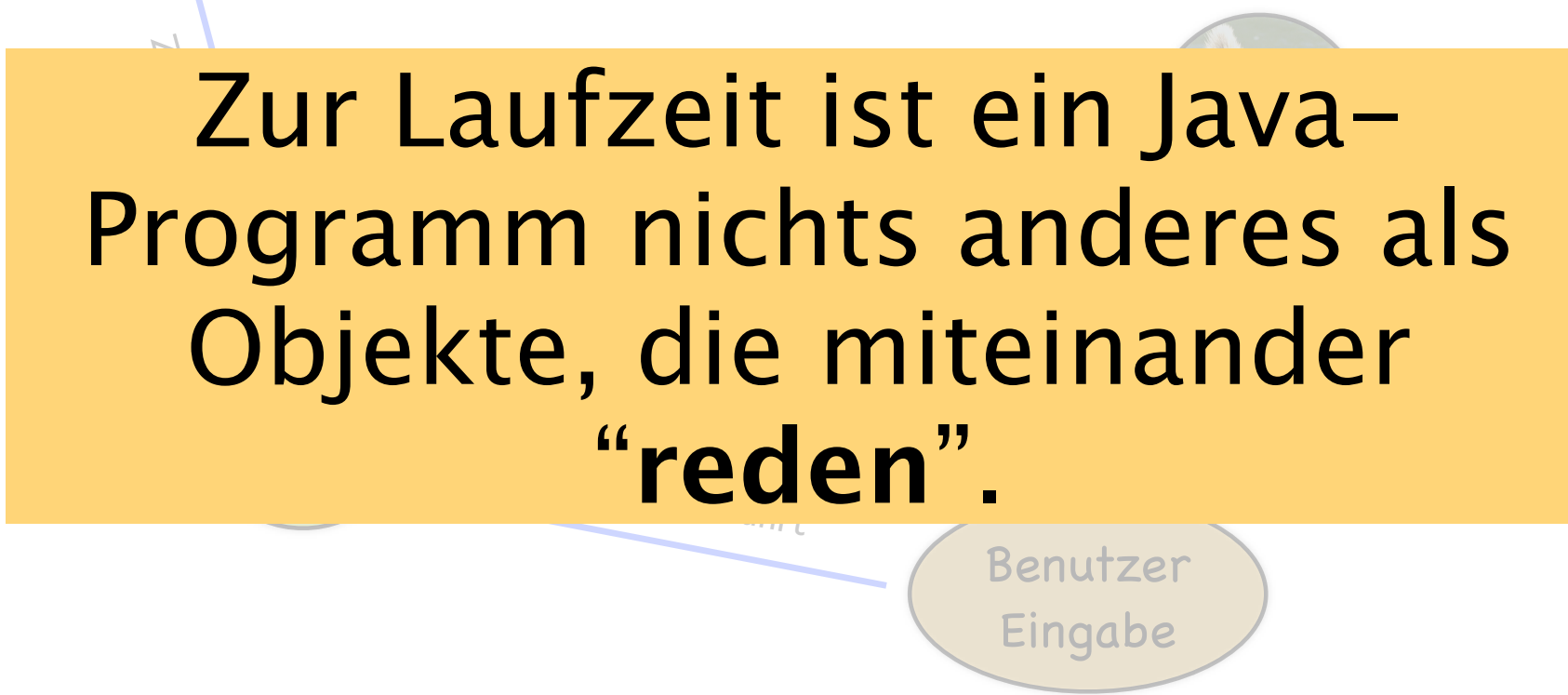
\$#!!!!\$!@#





Und die Moral von der Geschichte?

Zur Laufzeit ist ein Java-  
Programm nichts anderes als  
Objekte, die miteinander  
**“reden”**.



Benutzer  
Eingabe

Alle Objekte, die angelegt werden, kommen auf den Garbage Collectible **Heap**.

Ein Objekt erhält so viel Speicherplatz, wie es für seine Instanzvariablen benötigt.

Java merkt sich, welche Objekte Sie brauchen und welche nicht!

Wird der Speicher knapp, werden unbenutzte Objekte entsorgt!



Jetzt sind Sie  
wieder an der  
Reihe!



Schnappen Sie sich das Buch und lesen Sie  
**Kapitel 2.**