

Klassen und Objekte in der Objektorientierten Modellierung

Michael Pohlig



Oh, **Fifi** und
Fakir





Oh, zwei
Hunde





Oh, **Fifi** und
Fakir



Das Kind spricht von zwei Individuen



Oh, zwei
Hunde



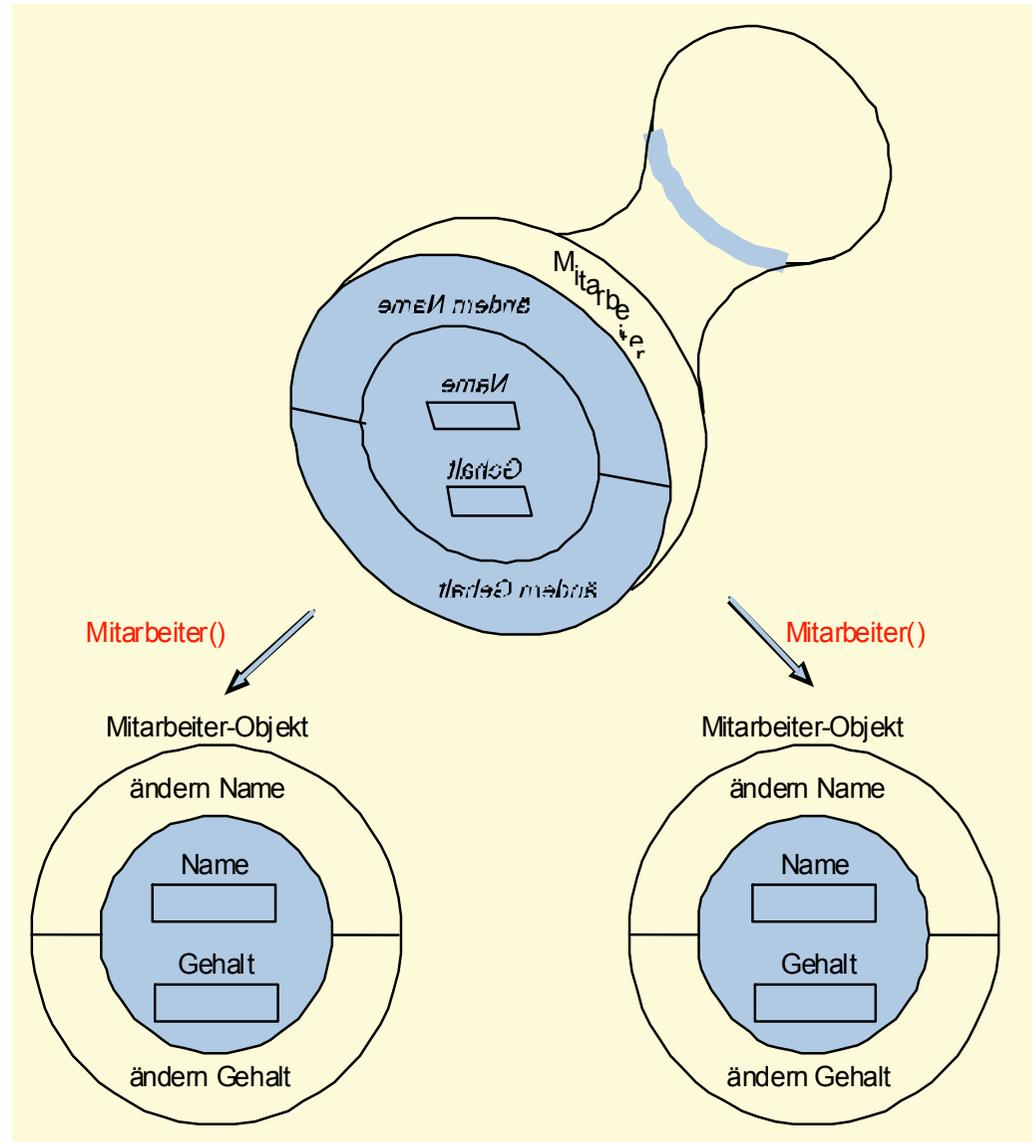
Das Kind kennt die Klasse Hund und erkennt die beiden Individuen als Instanzen oder Objekte der Klasse Hund.

Oder

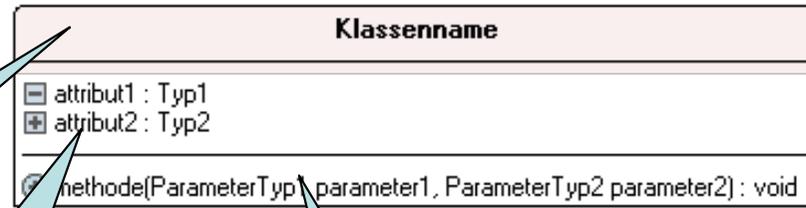
Objekte als Instanzen von
einer Klasse

Klasse: Punkt der Ebene

Objekt: Konkreter Punkt
P(2/4)



UML = Unified Modeling Language



1. Abteilung:
Name der Klasse

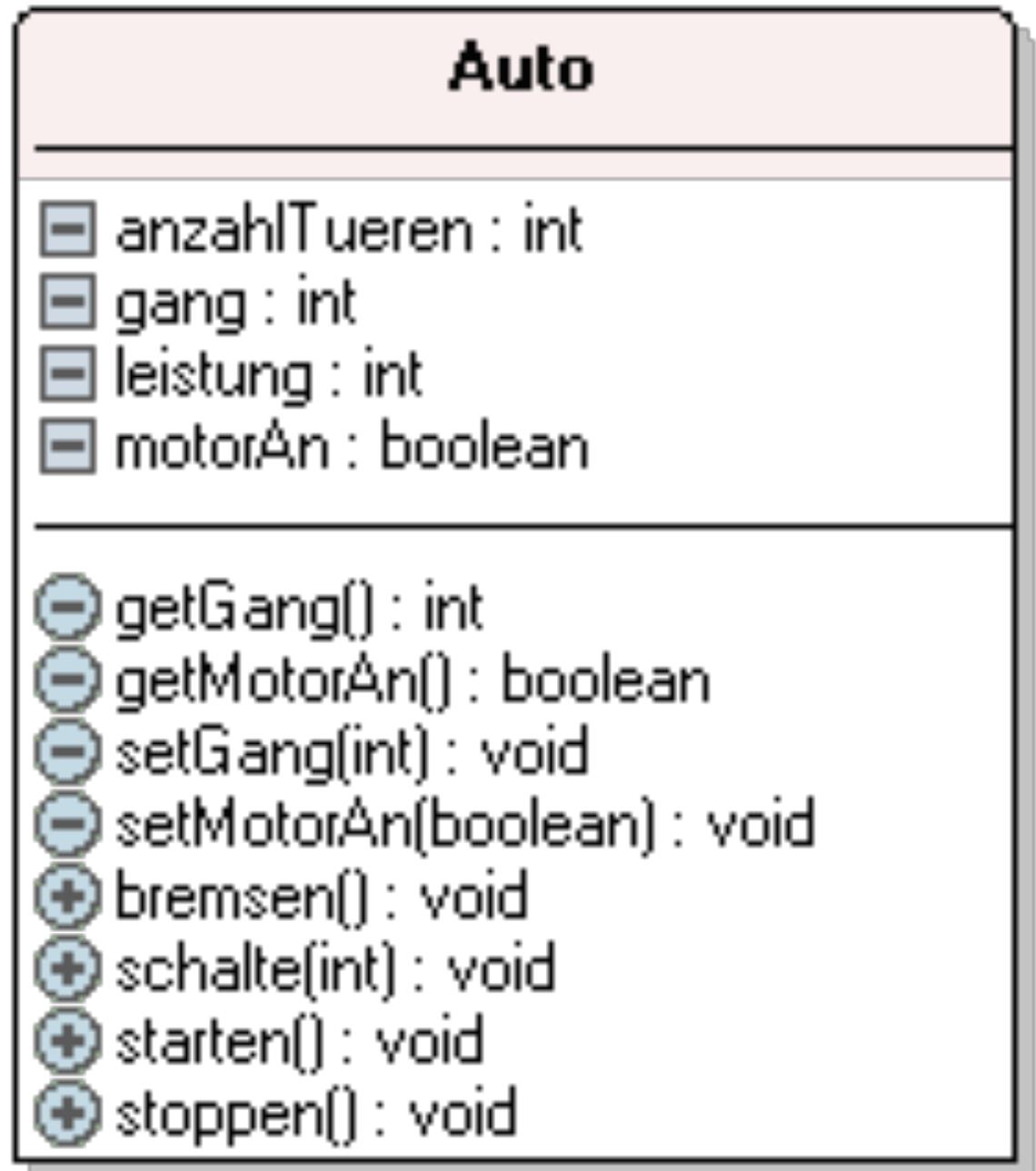
2. Abteilung: Attribute:
Modifizierer – Name des Attributes : Typ (int, double boolean, String...)

3. Abteilung: Methoden
Modifizierer – Name der Methode –
Parameterliste – Rückgabetypus

Kann leer sein, einen oder mehrere
Parameter haben

Void, oder Typ wie
double, boolean etc.

Wenn wir uns für den Kauf eines Autos interessieren, dann wollen wir wissen, was sein Motor leistet, wie viele Türen es hat, etc. Weiter erwarten wir, dass man ein Auto starten, anfahren, abbremsen, stoppen kann. Die Liste der Eigenschaften und der Methoden lässt sich fortsetzen.



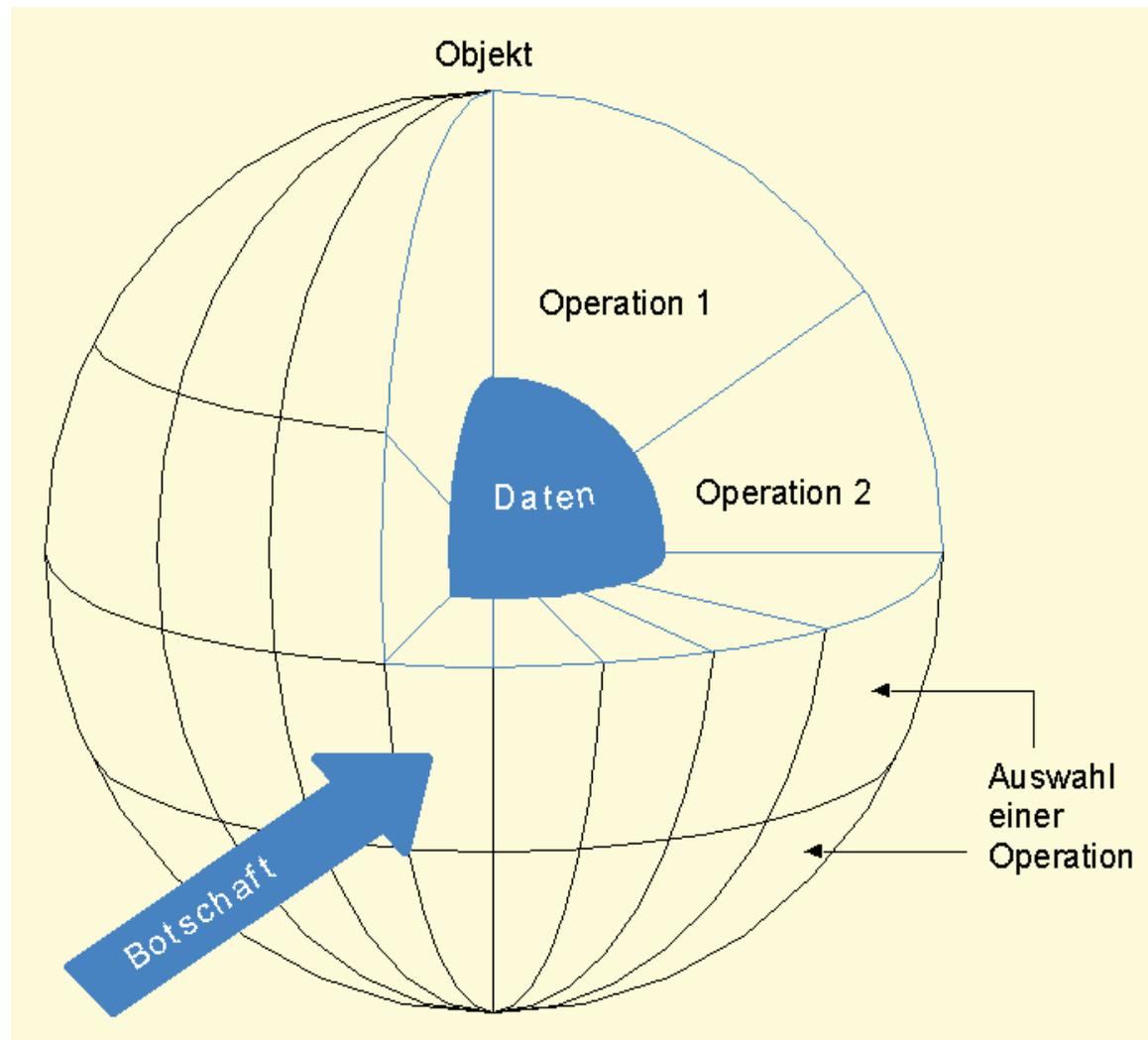
Auto

[-] anzahlTueren : int
[-] gang : int
[-] leistung : int
[-] motorAn : boolean

[-] getGang() : int
[-] getMotorAn() : boolean
[-] setGang(int) : void
[-] setMotorAn(boolean) : void
[+] bremsen() : void
[+] schalte(int) : void
[+] starten() : void
[+] stoppen() : void

```
meinAuto.setMotorAn(true);  
meinAuto.schalten(1);  
meinAuto.starten();  
meinAuto.schalten(2);  
meinAuto.schalten(3);  
meinAuto.bremsen();  
meinAuto.schalten(2);  
meinAuto.bremsen();  
meinAuto.stoppen();
```

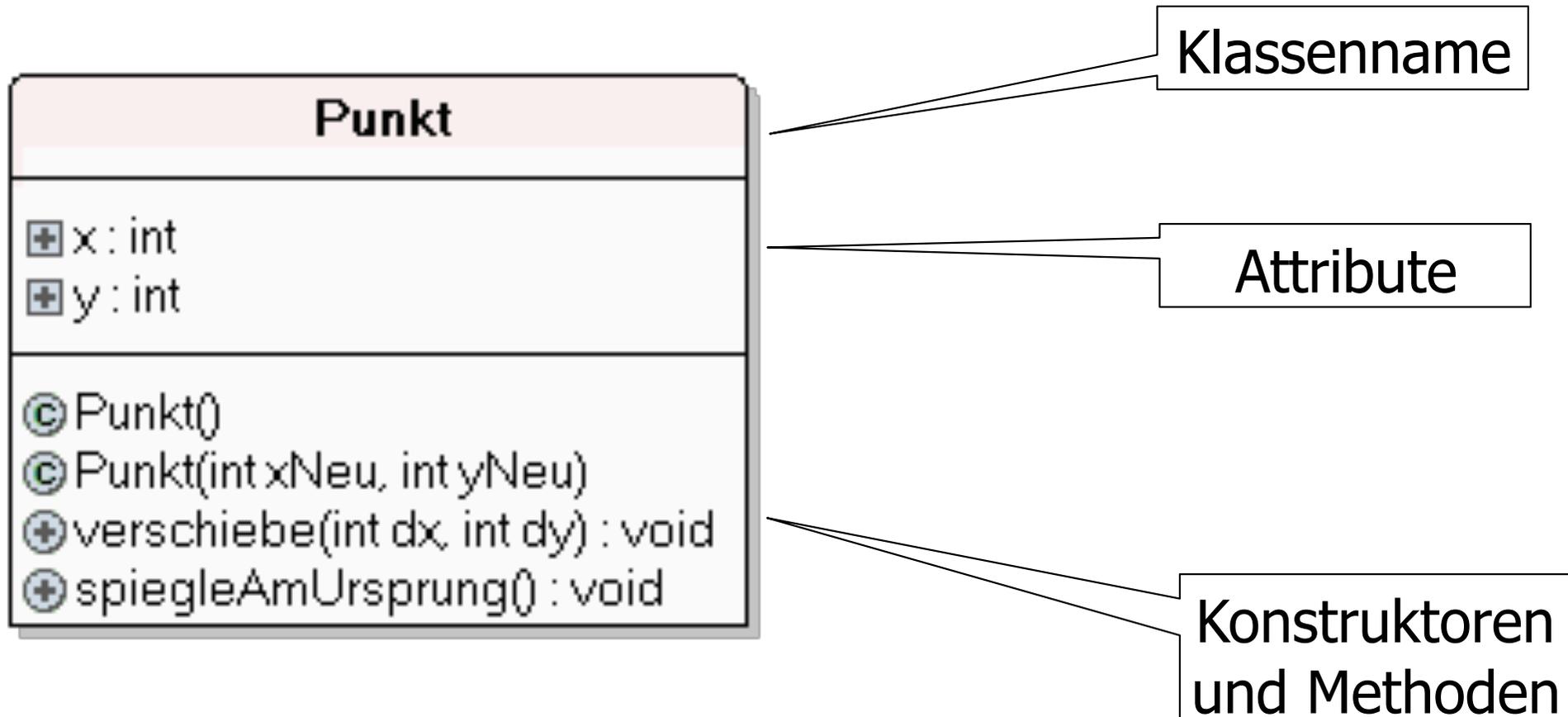
Information Hiding



private (-) schirmt ein Attribut nach außen ab.

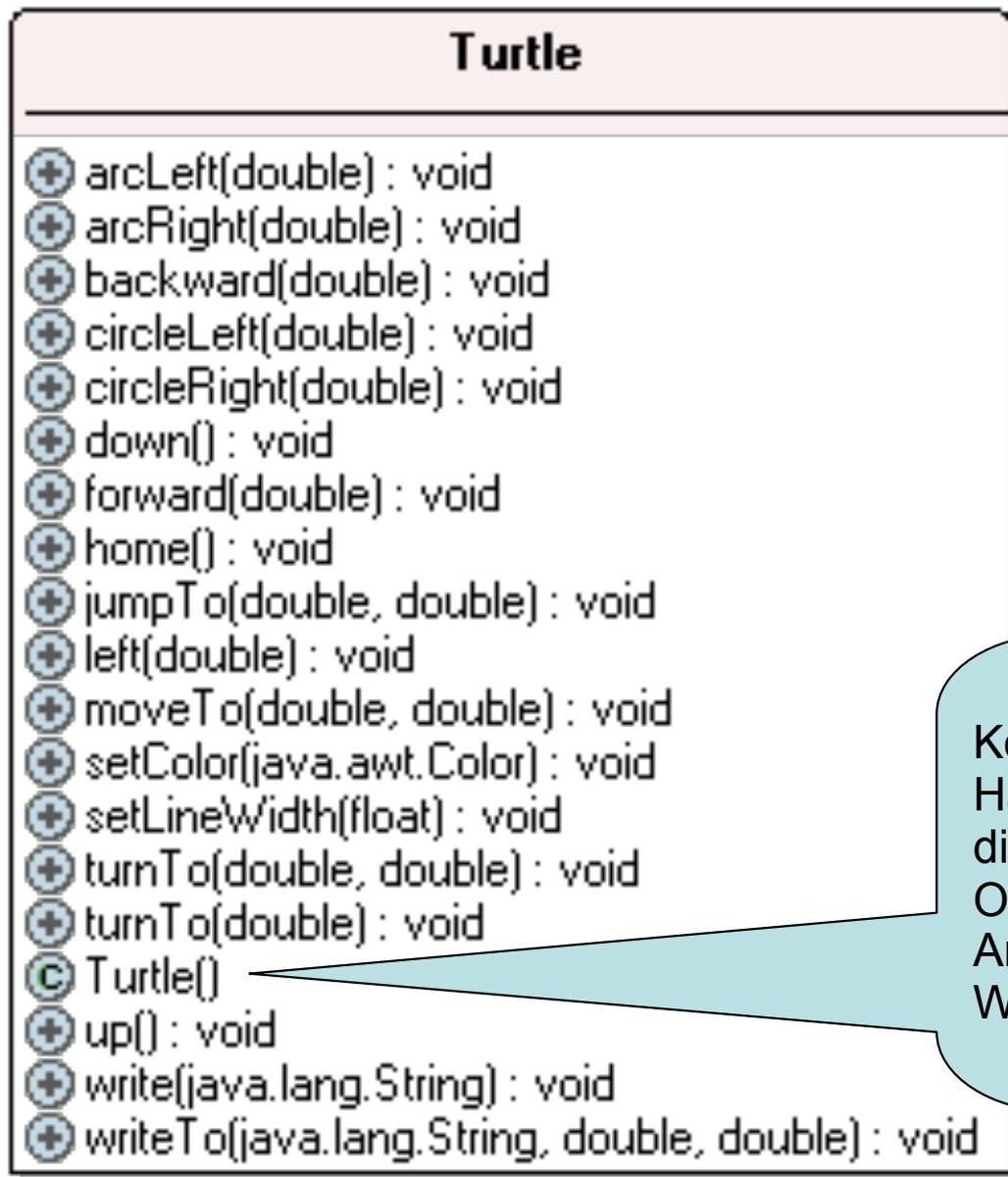
Methoden für den Zugriff auf die Daten

Eine Klasse bestimmt selbst, auf welche Attribute und auf welche Weise auf die Attribute seiner Instanzen zugegriffen werden darf bzw. kann.



```
class Punkt {
    int x,y;
    Punkt() {
        x=0;
        y=0;
    }
    Punkt(int xStartWert, int yStartWert) {
        x = xStartWert;
        y = yStartWert;
    }
    void verschiebe(int dx, int dy) {
        x +=dx;
        y +=dy;
    }
}
```

```
public class PunktDemo {  
    public static void main (String[] args) {  
        Punkt p1 = new Punkt();  
        Punkt p2 = new Punkt(3, 65);  
        //Ausgabe des Punktes P1  
        //Ausgabe des Punktes P2  
        p1.verschiebe(2, 2);  
        p2.verschiebe(2, 2);  
        //Ausgabe des Punktes P1  
        //Ausgabe des Punktes P2  
    }  
}
```



Konstruktor:
Hat den gleichen Namen wie die Klasse. Mit ihm werden Objekte erzeugt, die bestimmte Anfangseigenschaften haben. Welche?

Turtle

```

class IdealesGas {
    double T,V,S,p,E,n,...;
    finale double k = 1.38*10^(-23);
    IdealesGas() {
        T = 293; V = ...
    }
    idealesGas(double TStartWert,...) {
        T = TStartWert;
        ...
    }
    void komprimiereAdiabatisch(...) {
        .....
    }
}

```

Vererbung

