

Java:
Geheimnisprinzip, Kapselung;
Getter und Setter

Geheimnisprinzip

Nur einige Methoden und Attribute einer Klasse sind öffentlich zugänglich, die anderen bleiben geheim.

= weniger (unbefugte, ungewollte oder inkompetente) Eingriffsmöglichkeiten von außen (durch Mitprogrammierer, Böswillige ...)

Geheimnisprinzip; Kapselung

Beispiel Kundenkonto:

Öffentliche Methoden der Klasse:

- Kontostand abfragen
- Geld abheben

("von außen", bspw. am EC-Automat)

"Geheime" Methoden/Attribute der Klasse:

- Berechnung der Soll-/Habenzinsen
- Ermittlung der Kontoführungsgebühren
- Berechnung aktueller Kontostand



darf "nur" Kontostand
abfragen + Geld abheben

→

darf nicht
Kontostand berechnen,
Kontoführungsgebühren verändern usw.

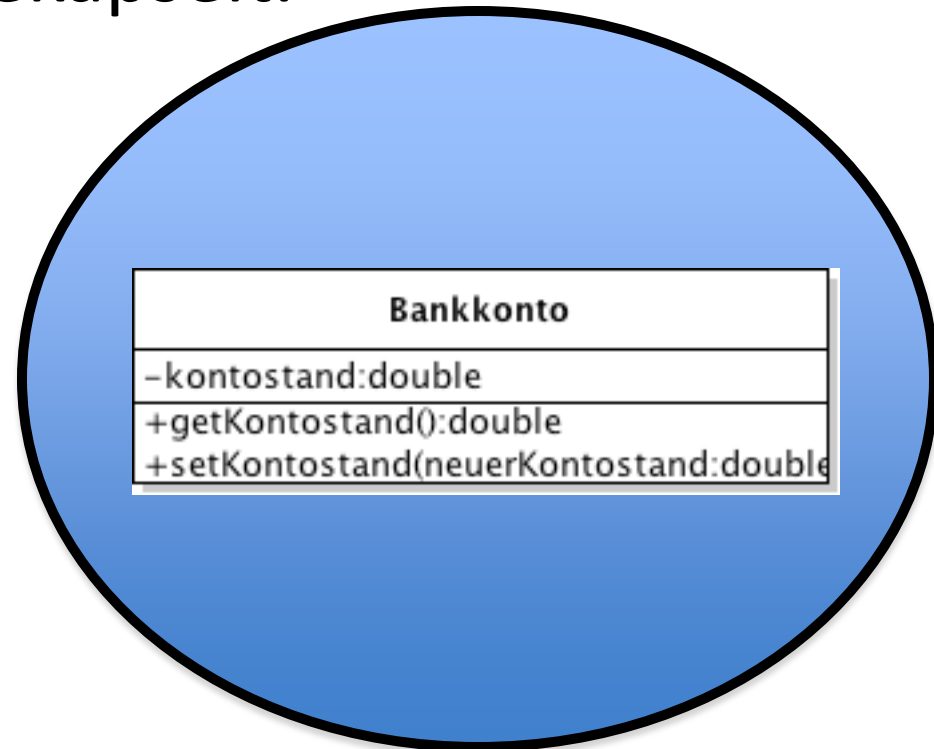
→



Kapselung

Methoden/Attribute, die nur über Zugriffsmethoden benutzt oder verändert werden können, sind gekapselt.

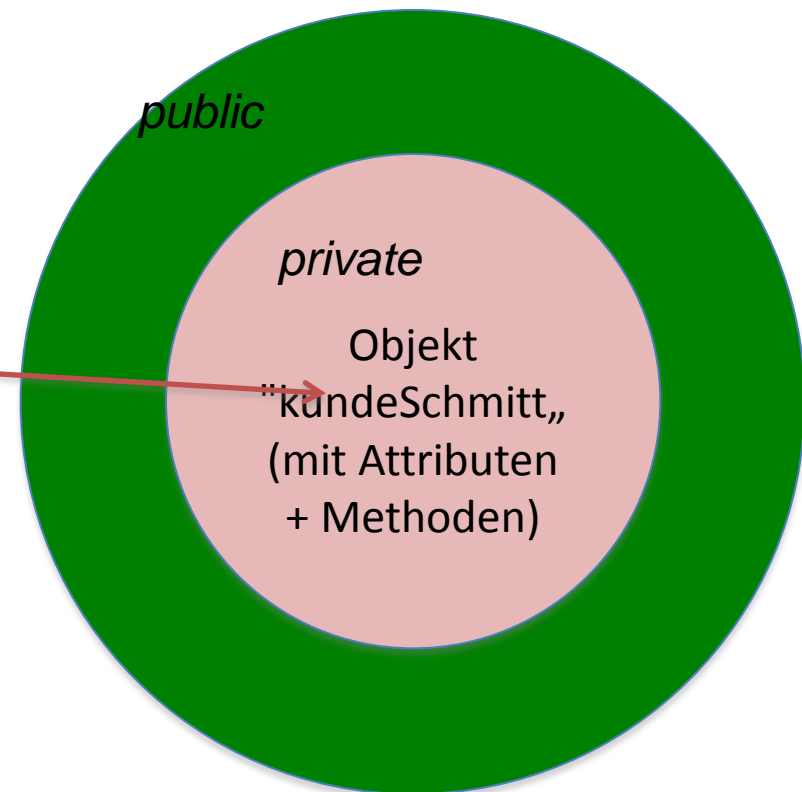
Attribut **kontostand**
kann nur über Getter-/
Setter-Methode abgefragt
oder verändert werden.



Kapselung

Zugriff auf private Attribute von außen NUR über Zugriffsmethoden möglich!

Zugriffsmethode (gewährt "Zugang" zu privaten Attr./Meth.)

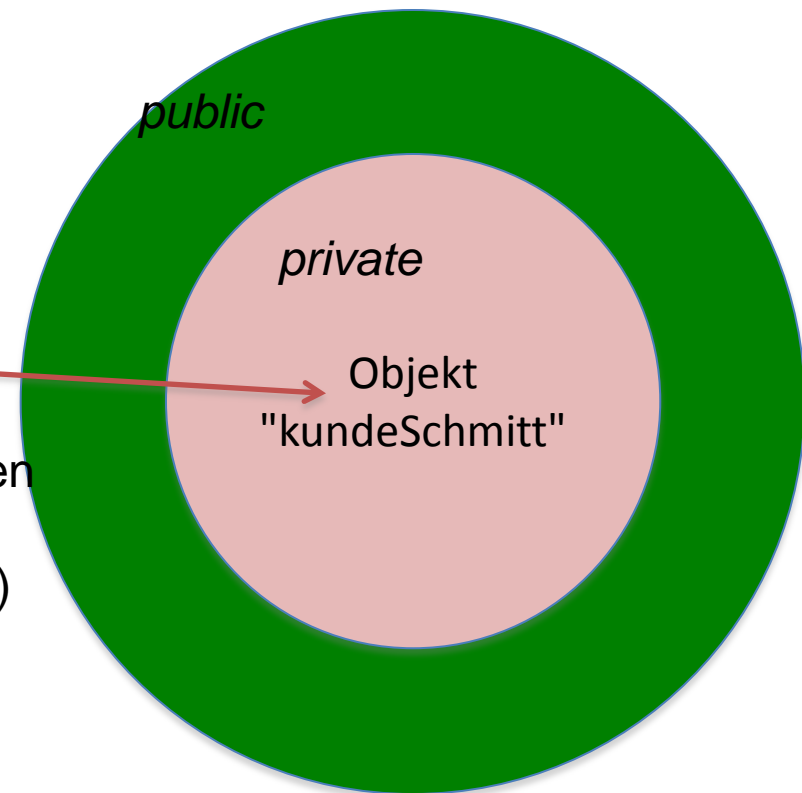


"Setter"

Methode, die die Veränderung einer Objektvariablen ermöglicht.

Verändern der Kundenadresse
über "Setter"-Methode

(aber keine Möglichkeit, Kontostand von außen
zu verändern, so lange keine
entsprechende Setter-Methode vorhanden ist)



"Setter"

```
public class Kunde
{
    // Attribute
    private double kontostand;
    // ...
    // Methoden
    public void setKontostand(double neuerKontostand){
        this.kontostand = neuerKontostand;
    }
    // ...
}
```

"Setter" - Eselsbrücke

Eine Hand greift in die Kapsel und weist einem Attribut einen neuen Wert zu.

Beim Hineingreifen:

→ in der Hand ist der neue Wert (= **Parameter**)

Beim Herausziehen:

→ Hand ist leer (= **kein Rückgabewert**)

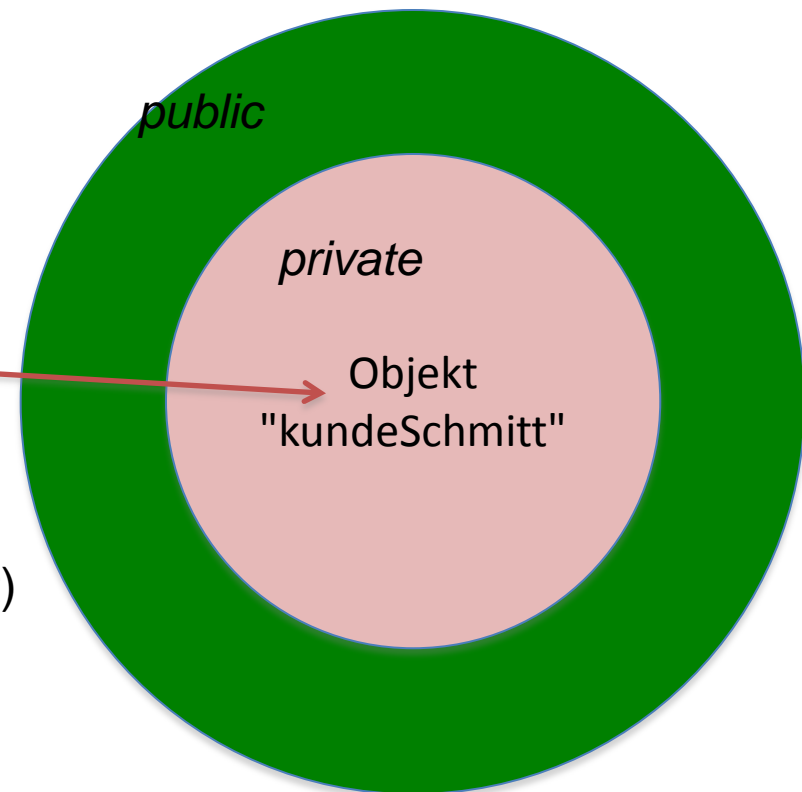
```
public void setAnzahl(int anzahl)
{
    return this.anzahl;
}
```


"Getter"

Methode, die eine Objektvariable abrufen.

Abrufen z.B. des Kontostandes mit
Getter-Methode von außen

(aber keine Möglichkeit, Kontostand von
außen zu abzurufen, so lange keine
entsprechende Getter-Methode vorhanden ist)



"Getter"

```
public class Kunde
{
    // Attribute
    private double kontostand = 130;
    // ...
    // Methoden
    public double getKontostand(){
        return this.kontostand;
    }
}
```

"Getter"

```
public class Kunde
{
    // Attribute
    private double kontostand = 130;
    // ...
    // Methoden
    public double getKontostand(){
        return this.kontostand;
    }
}
```

```
Kunde kundeSchmitt = new Kunde();
double schmittsKontostand = kundeSchmitt.getKontostand();
System.out.println("Herr Schmitt hat " + schmittsKontostand
+ " Euro auf dem Konto.");
```

Wie könnten Sie den obigen Code verkürzen?

"Getter" - Eselsbrücke

Eine Hand greift in die Kapsel und holt einen Wert.

Beim Hineingreifen:

→ Hand ist leer (= kein Parameter)

Beim Herausziehen:

→ in der Hand befindet sich ein Wert (= Rückgabewert)

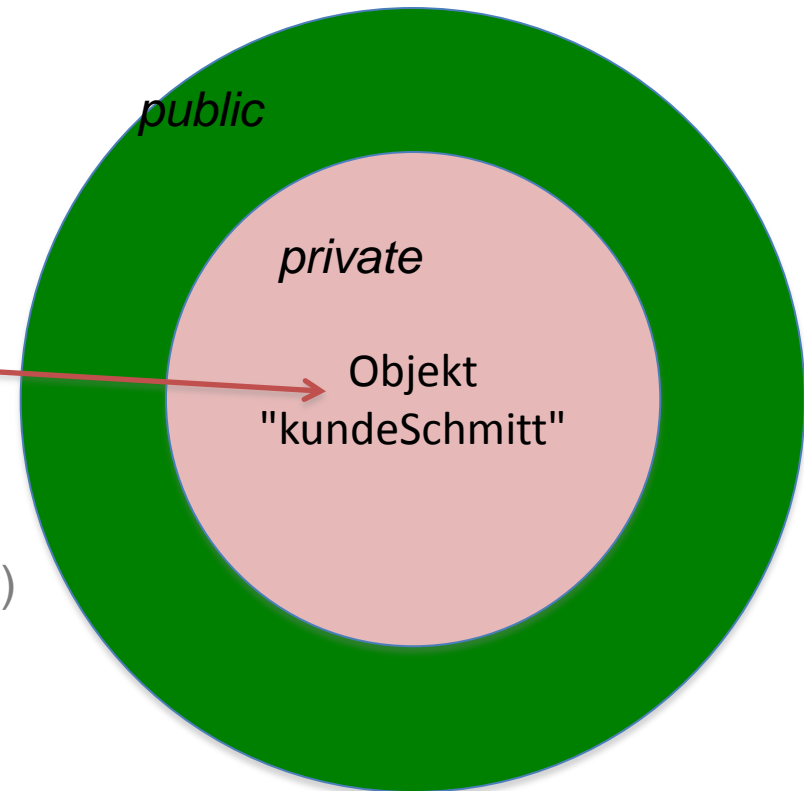
```
public int getAnzahl()  
{  
    return this.anzahl;  
}
```

Zugriff auf private Attribute unterbinden

Entfernen (oder private-Machen) der Zugriffsmethoden
= kein Zugriff von außen mehr möglich!

Abrufen z.B. des Kontostandes mit
Getter-Methode von außen

(aber keine Möglichkeit, Kontostand von
außen zu abzurufen, so lange keine
entsprechende Getter-Methode vorhanden ist)



Zusammenfassung: Kapselung

Attribute, Methoden sind in einer "Kapsel".

Zugriff nur über bestimmte Methoden möglich (z.B. Getter und Setter).

