

Kapselung

Wenn du eine Klasse nicht selbst programmiert hast sondern nur benutzen möchtest, dann interessieren dich nicht restlos alle Attribute und Methoden dieser Klasse sondern nur solche Methoden die dir helfen mit einem solchen Objekt arbeiten zu können.

Eine Klasse zu **kapseln** bedeutet, alles vor einem reinen Anwender zu verstecken, was er nicht unbedingt vom Aufbau der Klasse wissen muss. Im Gegensatz dazu bietet man einem reinen Anwender einen Satz von öffentlich zugänglichen Methoden an, die er verwenden kann. So stellst du als Programmierer sicher, dass ein fremder Benutzer deine Klasse nicht „falsch“ benutzt und dabei die Objekte „kaputt“ macht.

Modifikatoren

Vereinfacht gibt es die Modifikatoren **private** und **public**. Mit einem vorangestellten *private* kannst du den Zugriff auf ein Attribut oder auch eine Methode durch einen reinen Benutzer dieser Klasse verhindern. Mit einem vorangestellten *public* gewährst du einem reinen Benutzer Zugriff auf ein Attribut oder eine Methode.

Es gibt folgende **Faustregel für Programmier-Anfänger**:

- **Klassen sind immer public**
- **Attribute sind immer private**
Soll ein reiner Benutzer dennoch selbst einen Attribut-Wert ändern können, so bietet man ihm eine öffentliche (public) verändernde Methode zum Setzen dieses Attribut-Werts an.
Soll ein reiner Benutzer einen Attribut-Wert unmittelbar abfragen können, so bietet man ihm eine öffentliche sondierende Methode zum Nennen dieses Attribut-Werts an.
- **Methoden sind meist public**
Nur reine Hilfs-Methoden, deren Aufruf nicht von außerhalb der Klasse sinnvoll ist, macht man private.
- **Konstruktoren sind immer public**

Modifikatoren schreibt man beim Deklarieren eines Attributs oder einer Methode als erstes.

Code-Beispiel:

```
public class MENSCH
{
    private String name;
    private int alter;

    // jeder darf Objekte dieser Klasse erzeugen

    public MENSCH(String name)
    {
        this.name = name;
        this.alter = 0;
    }

    // jeder darf dich nach deinem Namen fragen

    public void nenneName()
    {
        return this.name;
    }

    // das möchtest du sicher nur selbst tun
    // und nicht von anderen erledigen lassen

    private void popoAbwischen()
    {
        // hier entsprechender Code (nicht Kot :-))
    }
}
```