

# Java

Kapselung über Zugriffsberechtigung (Zugriffsmodifizierer):

- Public (überall), private (nur innerhalb der Klasse), protected (nur innerhalb des Packages)
- static
- final (Konstante bzw in abgeleiteten Klassen ist Methode nicht überschreibbar)

## Klassen:

```
<Public> class <Klassenname> {Rumpf}
private class Auto { }
```

## Variablen:

```
<public> <Datentyp> <Name> = [Zuweisung];
private int leistung;
```

## Konstruktoren:

```
<public> <Klassenname> (Parameter) {Anweisung}
public Auto (int leistung, String hersteller) {this.leistung = leistung; (...) }
```

## Methoden:

```
<public> <Rückgabotyp> <Methodenname> (Parameter,Parameter...) { Anweisung}
```

## Konstanten:

```
Public static final int MAX = 100;
```

## Objekt:

```
<Klassenname> <Objektname> = new <Klassenname>(Parameter, falls Konstruktor welche
hat);
Obst Birne = new Obst (geld, 55);
```

## Import:

```
import <packagename>.*; (sämtliche Klassen)
import <packagename>.obst; (bestimmte Klasse)
import <Klassenname>; (falls Klasse public)
```

## Polymorphie:

Verwendung des gleichen Namens für unterschiedliche (aber miteinander verwandte) Dinge.  
Ermöglicht, dass verschiedenartige Objekte unter einem gemeinsamen Oberbegriff  
(Basisklasse) betrachtet und bearbeitet werden können.  
z.B.: Apfel und Birne sind beide Obst