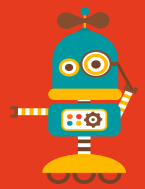
htl3 rennweg

Objektorientierte Programmierung (OOP)

DI Reinhold Buchinger

Höhere Abteilung für Mechatronik
Höhere Abteilung für Informationstechnologie
Fachschule für Informationstechnik



Lizenz/Credits

- Creative Commons-Lizenz CC BY-NC-SA 4.0 AT.
- Aufbauend auf den Folien von Ferdinand Kasper, (ferdinand.kasper@bildung.gv.at, Wien, Österreich)

Programmierparadigmen

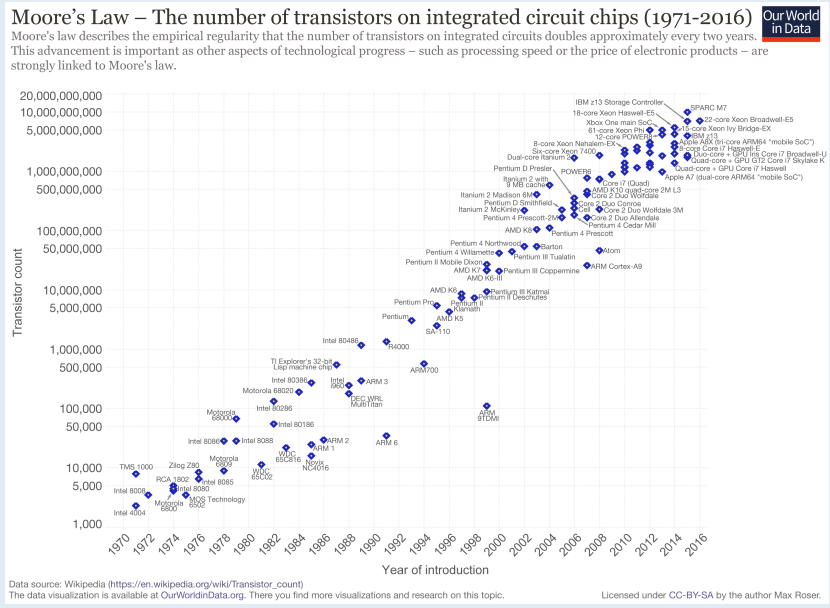
Programmierparadigmen

- Fundamentaler "Programmierstil"
- Programmiersprachen unterstützen meist mehrere Paradigmen, beruhen aber überwiegend auf einem.

Programmierparadigmen (Auswahl)	
Imperative Programmierung	Folge von Befehlen
Strukturierte Programmierung	Kontrollstrukturen (Schleifen, Bedingungen)
Prozedurale Programmierung	Programme werden in kleinere Teilaufgaben (Prozeduren) aufgeteilt.
Objektorientierte Programmierung (OOP)	Programm als eine Menge interagierender Objekte; Klassen, Vererbung, Polymorphie
Funktionale Programmierung	Alle Elemente können als Funktionen aufgefasst werden. Typische Einsatzgebiete: künstliche Intelligenz, Compilerbau, mathematische Anwendungen,..

Komplexität

- Leistungsfähigkeit von Computern steigt exponentiell
- Software-Entwickler/-innen bleiben immer ungefähr gleich schlau
- Ziel: Überblick behalten in immer komplexeren Programmen, neuere Programmierparadigmen helfen dabei.



Objekt und Klasse

Objektorientiere Programmierung (OOP)

htl3 rennweg
IT & MECHATRONIK

» OOP versucht Teile der (realen) Welt in einem Programm vereinfacht als Objekte abzubilden.

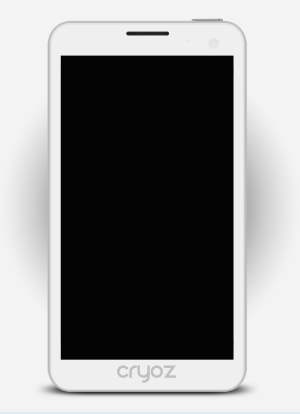
Jedes Objekt besitzt...

Eigenschaften/Merkmale/Zustände

Fähigkeiten/Verhalten

Beispiel Smartphone

Farbe
Gewicht
Betriebssystem
Version
Auflösung
Akkustand
gesperrt
....



Telefonieren
Nachricht senden
App installieren
App deinstallieren
Einschalten
....

Objektorientiere Programmierung (OOP)

htl3 rennweg
IT & MECHATRONIK

» OOP versucht Teile der (realen) Welt in einem Programm vereinfacht als Objekte abzubilden.

Jedes Objekt besitzt...

Eigenschaften/Merkmale/Zustände → Attribute

OOP Bezeichnungen

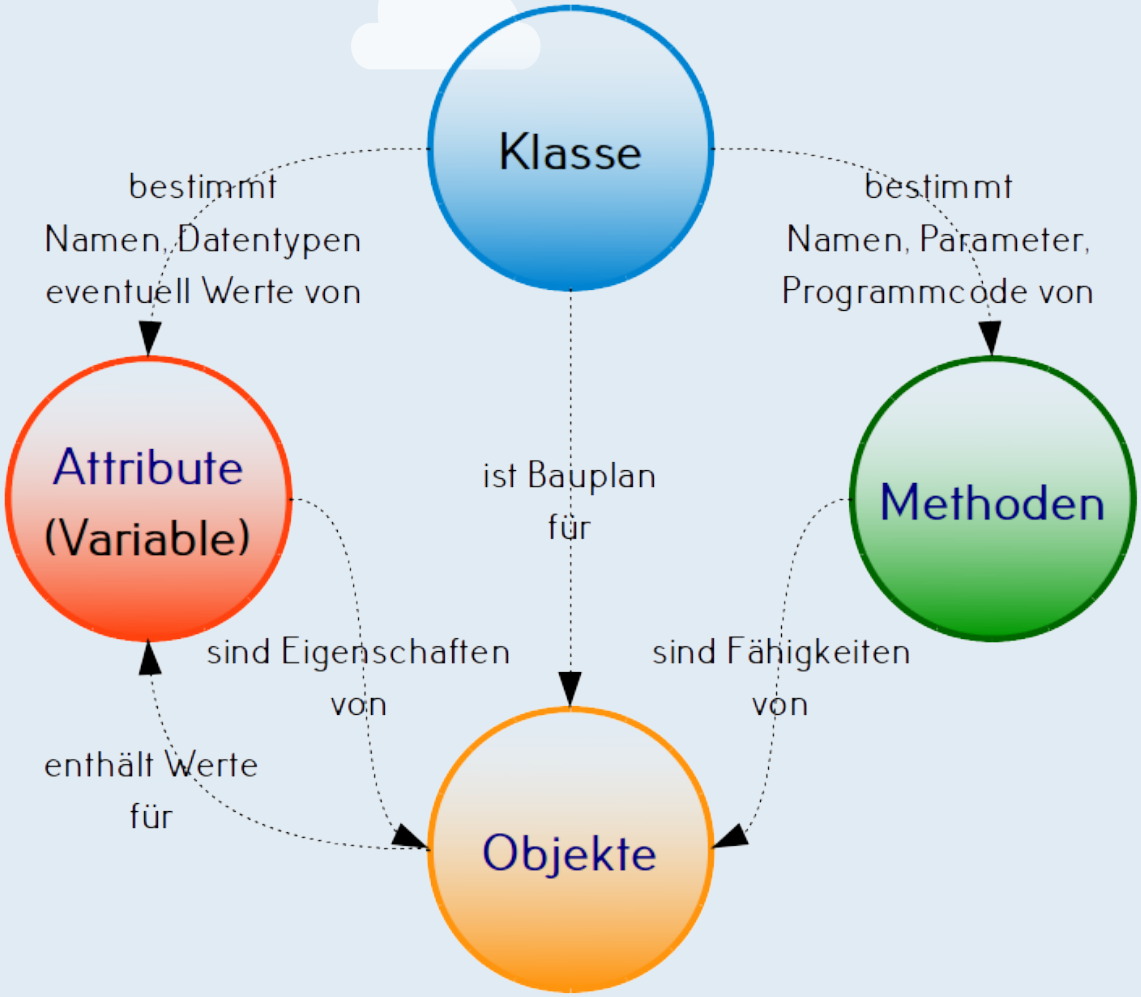
Fähigkeiten/Verhalten → Methoden

Klasse vs Objekt

htl3 rennweg
IT & MECHATRONIK

» Eine Klasse ist der **"Bauplan"** für Objekte.
» Von einer Klasse erzeugen wir Objekte.
» Ein Objekt ist eine konkrete Ausprägung einer Klasse.
» Beispiel Smartphone:

Zusammenhänge



OO-Analyse & OO-Design

- Wie findet man heraus, welche Klassen und Objekte man in einem Projekt braucht?

  OO-Analyse → OO-Design → OO-Programmierung

- **Objektorientierte Analyse (OOA)**
  - Fachleute beschreiben zunächst das Verhalten der Software nach außen hin
  - OOA „übersetzt“ diese Beschreibung in Klassen und Objekte aus dem Fachbereich
  - Man untersucht nur, was die Software tun soll (aber nicht, wie sie es tun soll)
- **Objektorientiertes Design (OOD)**
  - Legt fest, wie die Software ihre Aufgaben erfüllen soll
  - Mit Hilfe der Ergebnisse der OOA ermittelt das OOD die Klassen und Objekte für die spätere OOP.
- OOA und OOD werden in diesem Jahrgang nicht behandelt, sondern wir beschränken uns auf OOP

Klassen in Java

Klassen definieren

- Schlüsselworte **public class** → definiert eine neue Klasse
  - **public** wird später erklärt, vorläufig „Rezept“
  - Klassenname beginnt immer mit einem Großbuchstaben
  - Eine Klasse pro Datei → gleichnamige Quelltextdatei *.java

```
public class Smartphone {   // → Datei Smartphone.java
    // Attribute und Methoden von Personen
    ...
}

public class Bruch {   // → Datei Bruch.java
    // Attribute und Methoden von Bruch
    ...
}
```

Attribute definieren

- Attribut → Eigenschaft der Objekte einer Klasse
  - Entspricht einer Variablen innerhalb der Klasse → **Instanzvariable!**
  - Anfangswerte sind optional

```
public class Smartphone {
   String osSystem;     //Betriebssystem
   int gewicht;         //Gewicht in g
    ...
}

public class Bruch {
    int zaehler;        //Zähler
    int nenner = 1;         //Nenner mit Anfangswert
    ...
}
```

Methoden definieren

- Methode → Fähigkeit der Objekte einer Klasse
  - Für jede Fähigkeit → passende Methode innerhalb der Klasse

```
public class Smartphone {
    ...
    void klingle() {
        System.out.println("Beep Beep Beep Beep");
    }

}

public class Bruch {
    ...
    double alsKommazahl() {
        return (double) zaehler/nenner;
    }
}
```

Variablen & Objekte

Objekte erzeugen und speichern

- Operator **new** → liefert ein neues Objekt einer Klasse
- Neue Objekte können an beliebigen Programmstellen erzeugt werden.
- Klassenname wird zu einem Datentyp wie int, double, String usw.
- Klassenname = **Datentyp** für Variable, Parameter und Rückgabewerte

```
public class meinProgramm {

   public static void main(String[] args) {

      Bruch b = new Bruch();
      Bruch anotherBruch = new Bruch();

      Smartphone s1 = new Smartphone();
}
```

Referenzvariablen vs. primitive Variablen?

htl3 rennweg
IT & MECHATRONIK

- Wir haben Referenzvariablen bereits beim Thema Arrays im letzten Jahr kennen gelernt (siehe Unterlagen)
- Alle Variablen, die eine Klasse als Datentyp haben, sind Referenzvariablen.