

Prof. Dr Marcel Tilly

Objektorientiertes Programmieren

für Wirtschaftsinformatik

Objektorientierte Programmierung

aka: Programmieren 2

- Grundlegende Algorithmen und Datenstrukturen
- Einblick in die Objektorientierung
 - Modellierung und Programmierung
- Elemente professioneller Softwareentwicklung
- Format: 2 SWS Vorlesung, 2 SWS Übungen (Kleingruppen)
- Programmiersprache: Java (*heute: Javascript*)

Tagesmenü

- Kurzeinführung Javascript <repl.it/languages/javascript>
- Ein einfache Sortierverfahren:
 - *selection sort*: sortieren durch auswählen
 - *insertion sort*: sortieren durch einfügen
 - <https://visualgo.net/de/sorting>
- Vom Algorithmus zum Programm!

Über JavaScript ("JS")

- Skriptsprache von Netscape für dynamisches HTML in Webbrowsern (1995)
- Name entstand durch Kooperation mit Sun (Java)
- **JavaScript != Java**
- Unterstützt in allen gängigen Browsern
- Heute gibt es JavaScript in der Cloud (z.B. Serverless Functions), auf Devices (z.B. [Tessel Boards](#)) und Tools (z.B. Visual Studio Code)
- Ausführung ausserhalb des Browsers über [node.js](#)

Wer mitmachen möchte:

1. Browser öffnen
2. Gehe zu <https://jsbin.com/?js,console>

Kurzeinführung Javascript

- Ausgabe auf dem Bildschirm: `console.log()`
 - Text, aber in Anführungszeichen!
 - Variablen (siehe nächste Folie)
- Arithmetische Operationen: Grundrechenarten `+`, `-`, `*`, `/`,
`%`
 - `3 + 5`
 - `2 * -6`
 - Ergebnis: eine weitere Zahl
- Vergleichsoperationen: `<`, `>`, `<=`, `>=`, `==`, `!=`
 - `3 > 10`
 - `2 <= 2`
 - Ergebnis: ein Wahrheitswert

Kurzeinführung Javascript

- Variablen:
 - beliebige Bezeichner, keine Leer- oder Sonderzeichen
 - verwenden wie in Mathe:
 - `x = 10`
 - `i < 5`
 - `z = 2 + j`
 - Zuweisung: Variable muss immer links stehen, rechts ein Ausdruck

Kurzeinführung Javascript

- Felder (Arrays):
 - Mehrere Werte zusammengefasst unter einem Bezeichner
 - Wie ein Vektor in der Schulmathematik: mehrere "Dimensionen"
 - Zugriff (lesend wie schreibend) mit `[]` Operator, null-indiziert:
 - `a = []` : ein leeres Feld
 - `b = [1, 2, 3]` : ein Feld mit den drei Werten 1, 2 und 3
 - `b[1] = 10` : der **zweite** Wert in `b` ist jetzt 10

Kurzeinführung Javascript

- Bedingte Ausführung: `if - else if - else`
 - `if (b) { c }`
 - *lies:* wenn `b`, dann `c` ausführen

```
if (b[1] < b[2]) {  
    h = b[1]; b[1] = b[2]; b[2] = h  
}
```

- Wiederholte Ausführung:
 - `while (b) { c }`
 - *lies:* solange `b`, `c` ausführen
 - `for (i = 0; i < 10; i = i + 1) { }`
 - *lies:* für jeweils 0...9 als `i`, `c` ausführen

```
for(i = 0 ; i < b.length ; i++) {  
    console.log("Ergebnis " + b[i])  
}
```

Sortieren

Wie sortiert man eigentlich Spielkarten?

Sortieren durch Auswählen (Selection Sort)

Die Liste wird gedanklich in zwei Teile geteilt: einen sortierten (links) und einen noch unsortierten (rechts).

Zu Beginn ist die sortierte Liste leer.

Algorithmus

1. Suche das kleinste Element in der unsortierten Liste (Auswählen)
2. Tausche das kleinste mit dem ersten Element der unsortierten Liste
3. Das erste (kleinste) Element ist nun Teil der sortierten Liste.
4. Wiederhole bei 1.

```
a = [1, 5, 2, 3, 9, 8, 6, 7]
```

Implementierung

Selection Sort

```
a = [1, 5, 2, 3, 9, 8, 6, 7]
len = a.length

for (i = 0; i < len; i = i + 1) {
  p = i
  for (j = i+1; j < len; j = j + 1) {
    if (a[j] < a[p]) {
      p = j
    }
  }

  if (p != i) {
    hilf = a[i]
    a[i] = a[p]
    a[p] = hilf
  }
}

console.log(a)
```

Sortieren durch Einfügen (Insertion Sort)

Die Liste wird gedanklich in zwei Teile geteilt: einen sortierten (links) und einen noch unsortierten (rechts).

Zu beginn ist die sortierte Liste leer.

Algorithmus

1. Markiere das erste Element der unsortierten Liste als Kleinstes.
2. Vergleiche nun mit allen weiteren Elementen der unsortierten Liste
 - ist eines kleiner, so tausche mit dem Ersten.
3. Das erste (kleinste) Element ist nun Teil der sortierten Liste.
4. Wiederhole bei 1.

```
a = [1, 5, 2, 3, 9, 8, 6, 7]
```

Implementierung

Insertion Sort

```
a = [1, 5, 2, 3, 9, 8, 6, 7]
len = a.length

for (i = 0; i < len; i = i + 1) {
  for (j = i+1; j < len; j = j + 1) {
    if (a[j] < a[i]) {
      hilf = a[i]
      a[i] = a[j]
      a[j] = hilf
    }
  }
}

console.log(a)
```