

# Programmiermethoden und -werkzeuge 1

*Woche 1 - Einführung*

**Jochen Hosenfeld**

*jochen.hosenfeld@informatik.hs-fulda.de*

*Fachbereich Angewandte Informatik*

October 17, 2025

# Studentische Partizipation

AstA

Fachschaftsrat

Instagram

# Programmierwerkzeuge

# Programmierwerkzeuge

Brainstorming-Aufgabe: Was sind essenzielle Programmierwerkzeuge?

**DON'T LOSE**

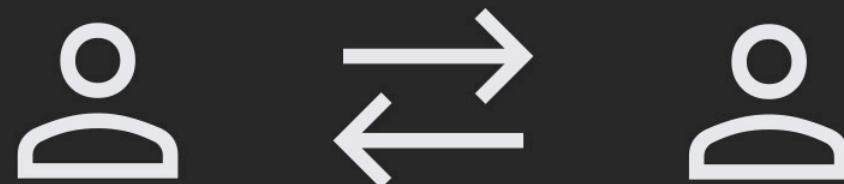
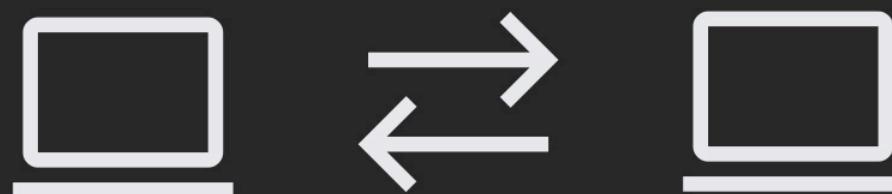
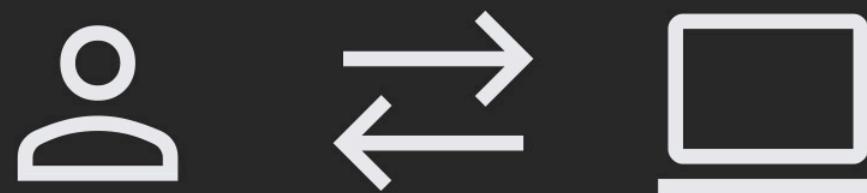


**YOUR HEAD**

# Beratung

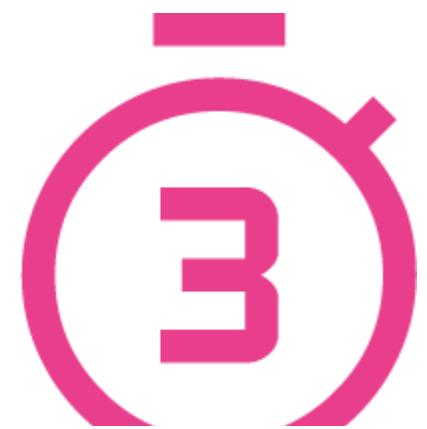
## Beratungsangebote

# Kommunikation



# Kommunikation

Partneraufgabe: Was braucht es, damit Kommunikation gelingt?



# Kommunikation

Partneraufgabe: *Was braucht es, damit Kommunikation gelingt?*

- Austausch oder Übertragung von Informationen
- Präzise und verständlich
- Logischer Aufbau
- Zielgerichtet
- Der Zielgruppe angepasst
- Feedback
- Ehrlichkeit, Offenheit, ...

# Kommunikation

Was ist ein ...

- Parameter?
- Argument?
- Attribut?



**Gemeinsames Vokabular**

# Programmiermethoden

# Methoden

Partneraufgabe: Was ist eine Methode?



# Methoden

Partneraufgabe: Was ist eine Methode?

## Cambridge-Dictionary

„eine Art und Weise, etwas zu tun, oft eine, die ein System oder einen Plan beinhaltet“

## Duden

„auf einem Regelsystem aufbauendes Verfahren zur Erlangung von [wissenschaftlichen] Erkenntnissen oder praktischen Ergebnissen“

# Methoden

Wenn eine Methode ...

- gezielt eine Aufgabe löst,
- dabei klar definierten Schritten folgt,
- die auf festen Regeln basieren,
- und sie nach einer bestimmten Anzahl von Schritten zu Ende ist,

dann ist sie ein **Algorithmus**.

# Fragen zu Methoden

Ist jede Methode ein Algorithmus? **Nein**

Was wäre ein Beispiel für eine Methode, die kein Algorithmus ist? **Lehre**

Warum? **Zielorientiert ohne klar definierte Schritte**

Woher weiß ich, was die Ziele des Moduls sind? **Prüfungsordnung**

# Portfolio

## Allgemeine Bestimmungen für Studien- und Prüfungsordnungen

„Bei einem Portfolio ist eine Leistungssammelmappe zu erstellen, die den Lernprozess der zu prüfenden Person durch Zusammenstellung geeigneter kleinerer Texte oder Daten, Recherchen oder Hausaufgaben, Artikel und ähnlicher Materialien sowie durch einen Selbstreflexions- bericht dokumentiert.“

# Portfolio-Abgabe

- 12 Themen dokumentieren und reflektieren
- Bewertungskriterien: Inhaltliche und formale Korrektheit, Struktur
- Portfolio in GitLab erstellen und einreichen (Anleitung wird in den nächsten Wochen behandelt)
- Abgabe am Ende der Vorlesungszeit
- Bei Fragen während der Übungen oder in der Sprechstunde melden

# Weitere Schlüsselkompetenzen

Lernkompetenz

Selbständigkeit und Eigeninitiative

Soziale Kompetenz

Kommunikative Kompetenz

Digitale Kompetenz

# Programmiermethoden und - werkzeuge 1

Wie ein Programmierer  
denken zu können ...

# ***METHODEN***

... wie ein Programmierer  
handeln zu können

## **WERKZEUGE**

# Repräsentation

# Binärsystem

Das Binärsystem funktioniert auf der Basis von zwei Ziffern: **0** und **1**.

Jedes Bit in einer binären Zahl repräsentiert eine Potenz von **2**, mit dem niedrigsten Exponenten beginnend an der rechten Seite.

# Binärsystem

Betrachten wir die binäre Zahl **101**:

- Rechtes Bit steht für  **$2^0$** , multipliziert mit dem Wert des Bits (1) => **1**
- Mittleres Bit steht für  **$2^1$** , multipliziert mit dem Wert des Bits (0) => **0**
- Linkes Bit steht für  **$2^2$** , multipliziert mit dem Wert des Bits (1) => **4**

Addiert man diese Werte ( **$4 + 0 + 1$** ), erhält man den Dezimalwert **5**.

# Dezimalsystem

Betrachten wir die Dezimalzahl **101**:

- Rechtes Bit steht für  **$10^0$** , multipliziert mit dem Wert des Bits (1) => **1**
- Mittleres Bit steht für  **$10^1$** , multipliziert mit dem Wert des Bits (0) => **0**
- Linkes Bit steht für  **$10^2$** , multipliziert mit dem Wert des Bits (1) => **100**

Addiert man diese Werte ( **$100 + 0 + 1$** ), erhält man den Dezimalwert **101**.

# Hexadezimalsystem

- Basis 16
- Zahlen 0-9, Buchstaben A-F
- RGB-Farbcodes:
  - #RRGGBB in Web-Design
  - 0xRRGGBB in Programmiersprachen
- Farbdarstellung: 16 Millionen Möglichkeiten

# ASCII

## American Standard Code for Information Interchange

ASCII Wert	Zeichen								
32	Space	51	3	70	F	89	Y	108	I
33	!	52	4	71	G	90	Z	109	m
34	"	53	5	72	H	91	[	110	n
35	#	54	6	73	I	92	\	111	o
36	\$	55	7	74	J	93	]	112	p
37	%	56	8	75	K	94	^	113	q
38	&	57	9	76	L	95	_	114	r
39	'	58	:	77	M	96	`	115	s
40	(	59	;	78	N	97	a	116	t
41	)	60	<	79	O	98	b	117	u
42	*	61	=	80	P	99	c	118	v
43	+	62	>	81	Q	100	d	119	w
44	,	63	?	82	R	101	e	120	x
45	-	64	@	83	S	102	f	121	y
46	.	65	A	84	T	103	g	122	z
47	/	66	B	85	U	104	h	123	{
48	0	67	C	86	V	105	i	124	
49	1	68	D	87	W	106	j	125	}
50	2	69	E	88	X	107	k	126	~

# UTF-8

## *Unicode Transformation Format – 8-bit*

- Universelles Zeichenset für moderne und ausgestorbene Sprachen
- Rückwärtskompatibel mit ASCII
- Variabler Zeichenbreiten-Standard (1 bis 4 Bytes)
- Selbstkorrekturmecanismus

# Markup

- Auszeichnungssprache
- Textinhalte ausgezeichnet, formatiert, gegliedert
- „Trennung von Struktur und Darstellung“
- HTML (Hypertext Markup Language) für Webseiten
- XML (Extensible Markup Language) für Datenaustausch
- LaTeX für wissenschaftliche Dokumente

# HTML

## I am a Subheadline

- I am **strong**
- I am *italic*
- And I am `code: i = 1`

```
1 <h1>HTML</h1>
2 <h2>I am a Subheadline</h2>
3
4 <ul>
5   <li>I am <strong>strong</strong></li>
6   <li>I am <em>italic</em></li>
7   <li>And I am <code>code: i = 1</code></li>
8 </ul>
```

# Markdown

## I am a Subheadline

- I am **strong**
- I am *italic*
- And I am `x = 1`

```
1 # Markdown
2 ## I am a Subheadline
3
4 - I am **strong**
5 - I am *italic*
6 - And I am `code: x = 1`
```

# Markdown vs. HTML

```
1 # Markdown
2 ## I am a Subheadline
3
4 - I am **strong**
5 - I am *italic*
6 - And I am `code: x = 1`
```

```
1 <h1>HTML</h1>
2 <h2>I am a Subheadline</h2>
3
4 <ul>
5   <li>I am <strong>strong</strong></li>
6   <li>I am <em>italic</em></li>
7   <li>And I am <code>code: i = 1</code></li>
8 </ul>
```

# Erfolg von Markdown schafft Probleme

- Informelle Anleitung anstatt präzise Spezifikation
- Fehlende Features
- Vielzahl von Implementationen -> Unklarheiten

# CommonMark

- Initiative zur Standardisierung
- Konsistenz
- Kompatibilität
- Aktive Community

# Themen von heute

- Organisatorisches
- Formen der Kommunikation
- Methode und Algorithmus
- Repräsentation von Zahlen, Buchstaben und Farben
- Markdown und CommonMark

Wie ein Programmierer zu  
denken und zu handeln