

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>15</b>
<b>1 Intuitive Modelle und informatisches Wissen</b>	<b>17</b>
1.1 <i>Intuition als Wissen</i>	17
1.2 <i>Intuition als prozedurales Wissen</i>	17
1.3 <i>Intuitive Modelle als bedeutungsbezogenes deklaratives Wissen</i>	18
1.3.1 Begriffe – Identifikation von Objekten	18
1.3.2 Schemata – Speichern struktureller Merkmale	18
1.3.3 Mentale Modelle	19
1.4 <i>Merkmale intuitiver Modelle</i>	19
1.5 <i>Phänomenologische Primitive</i>	21
1.6 <i>Intuitive Modelle und Fehlvorstellungen</i>	21
<b>2 Repräsentation intuitiver Modelle</b>	<b>23</b>
2.1 <i>Duale Kodierung von Wissen</i>	23
2.2 <i>Vielfalt und Flüchtigkeit</i>	24
2.3 <i>Repräsentation und Metaphorisierung</i>	24
2.3.1 Uneigentliche Redeweise in der Informatik	24
2.3.2 Metaphern und analoges Denken	25
2.3.3 Metaphern als Bilder für intuitive Modelle	26
2.4 <i>Beispiele als Repräsentationen intuitiver Modelle</i>	28
2.4.1 Prinzipien der Beispielbildung	28
2.4.2 Prototypische Beispiele	29
2.4.3 Repräsentation durch eine Beispielkollektion	29
2.4.4 Beispiel-basiertes Problemlösen	30
2.5 <i>Strukturorientierte und ablauforientierte Repräsentationen</i>	30
2.6 <i>Modellrepräsentationen im Unterricht</i>	31
<b>3 Verwendung intuitiver Modelle</b>	<b>33</b>
3.1 <i>Verstehen</i>	33
3.1.1 Verstehen aus Sicht der Hermeneutik	34
3.1.2 Verstehen durch intuitive Modelle	34
3.2 <i>Erklären</i>	35
3.2.1 Fokussierung	35
3.2.2 Mehrperspektivität: Viele Modelle für eine Sache	37
3.3 <i>Problemlösen</i>	37
3.3.1 Antizipatorische Intuitionen	37
3.3.2 Ansatz und Verfeinerung	38
3.3.3 Paradigmatische Modelle und Software-Entwicklung	39
3.3.4 Entwurfsmuster (Design Patterns)	40
3.3.5 Use Cases – intuitive Modelle für Funktionalität	40
3.3.6 Intuitive Modelle im agilen Programmieren – The Planning Game	41
3.4 <i>Kontrollmodelle</i>	41
3.4.1 Zusicherungen	42
<b>4 Empirische Erforschung intuitiver Modelle</b>	<b>45</b>
4.1 <i>Forschungsansätze und methodische Probleme</i>	45
4.2 <i>Visualisierungsübungen</i>	45
4.3 <i>Die Python Visual Sandbox</i>	46

4.3.1	Die PVS als Spiel	46
4.3.2	Lernen mit der PVS	47
4.3.3	Überblick über den technischen Aufbau der Python Visual Sandbox	47
4.4	<i>Python Visual</i>	48
4.5	<i>Python Puzzle</i>	49
4.5.1	Dokumentation einer Session	51
4.5.2	Auswertung der Python Puzzle Sessions	51
4.5.3	Verwendung von Tipps	52
4.6	<i>Python Puzzle assert</i>	53
4.7	<i>Python Quiz</i>	53
4.7.1	Bewertung der Antworten	54
4.7.2	Dokumentation einer Session	54
4.7.3	Schlussfolgerungen zur Intuitivität von Modellen	55
4.8	<i>Workshops mit der PVS</i>	55
4.9	<i>Systematisierung intuitiver Modelle der Informatik</i>	56
<b>5</b>	<b>Akteurmodelle</b>	<b>57</b>
5.1	<i>Daten als Akteure – Datenflüsse</i>	57
5.2	<i>Namen als Akteure</i>	58
5.3	<i>Funktionen</i>	58
5.4	<i>Allmächtige Steuerungsentität – monoaktive Systeme</i>	58
5.5	<i>Objekte</i>	59
<b>6</b>	<b>Benennung von Entitäten</b>	<b>61</b>
6.1	<i>Behältermodell und Referenzierungsmodell</i>	61
6.2	<i>Erscheinungsmodelle</i>	62
6.3	<i>Wem gehört ein Name? Zeiger versus Etiketten</i>	64
6.4	<i>Vermischung von Namensmodellen</i>	65
6.5	<i>Namen als Bezeichnungen für Rollenträger</i>	66
6.6	<i>Implizite Namen</i>	66
6.7	<i>Indirekte Namen</i>	68
6.7.1	Funktionsaufrufe und mathematische Terme	68
6.7.2	Benennung durch Literale	69
6.8	<i>Assoziationen</i>	69
6.9	<i>Benennung als Unterordnung</i>	70
<b>7</b>	<b>Daten</b>	<b>73</b>
7.1	<i>Ansicht versus Literal</i>	73
7.2	<i>Verwechseln von Wert (Datum) und Literal</i>	73
7.3	<i>Figürliche Ansichten</i>	73
7.4	<i>Nichts</i>	74
7.5	<i>Platzhalter für variable Teile in Dokumenten</i>	75
7.6	<i>Daten als Entitäten oder Zustände von Objekten</i>	76
<b>8</b>	<b>Funktionen</b>	<b>79</b>
8.1	<i>Funktion als Box mit Ein- und Ausgang für Daten</i>	80
8.2	<i>Dateneingabe über „Sensoren“</i>	81

8.3	<i>Übergabe von Referenzen bei der Eingabe</i>	82
8.4	<i>Ursprung der Eingabespezifikation</i>	82
8.5	<i>Vergleich von Eingabemechanismen</i>	82
8.6	<i>Übergabe von Referenzen bei der Ausgabe</i>	84
8.7	<i>Ausgabe als Signal</i>	86
8.8	<i>Durchlässigkeit der Systemgrenze</i>	87
8.8.1	Geschlossene Box	87
8.8.2	Box mit „Seitentür“	87
8.8.3	Direkter Zugriff auf externe Objekte	87
8.8.4	Vergleich von offenen und geschlossenen Funktionsmodellen	88
8.9	<i>Dynamische und statische Funktionsmodelle</i>	89
8.10	<i>Auslösemechanismen</i>	91
<b>9</b>	<b>Kontrolle – Steuerung</b>	<b>93</b>
9.1	<i>Handhabung von Kontrolle: Kontrollfluss und Kontrollübergabe</i>	93
9.2	<i>Anweisungssequenzen</i>	93
9.3	<i>Bedingte Anweisungen</i>	94
9.3.1	Verzweigung des Kontrollflusses	94
9.3.2	Kontrolle von Datenflüssen – Datenweichen und Datensperren	95
9.3.3	Ereignismodell – Steuersignale	95
9.4	<i>Iterationen – datengesteuerte Wiederholungen</i>	95
9.5	<i>Wiederholungen mit nicht antizipierbarem Ende</i>	97
9.5.1	Kontrollierte Wiederholung einer holistischen Aktivität	98
9.5.2	Schleifen	99
9.6	<i>Rekursion</i>	99
9.6.1	Rekursion als Schleife	100
9.6.2	Rekursion als Selbstaufforderung	100
9.6.3	Fehlerhafte Verwendung des Modells der Selbstaufforderung	101
9.6.4	Delegationsmodell	102
9.6.5	Protokoll-Modelle für rekursive Algorithmen	102
9.6.6	Schema einer rekursiven Funktion und Dedynamisierung	102
<b>10</b>	<b>Verarbeitung</b>	<b>105</b>
10.1	<i>Entstehen</i>	105
10.1.1	Entstehen von Daten	105
10.1.2	Entstehung von Namen	105
10.2	<i>Vernichtung</i>	105
10.2.1	Implizite Vernichtung bei Zuweisungen	105
10.2.2	Sukzessive Zuweisungen ohne Vernichtung	107
10.2.3	Totaler Vernichtung	108
10.3	<i>Veränderung (Metamorphose)</i>	108
10.3.1	Datenumwandlungen	109
10.4	<i>Namenumwandlungen</i>	109
10.5	<i>Bewegen</i>	110
10.5.1	Bewegung von Daten	110
10.5.2	Modellierung von Zuweisungen durch Datenbewegung	111
10.5.3	Modellierung von Zuweisungen durch Namenbewegungen	112
<b>11</b>	<b>Klassen</b>	<b>117</b>
11.1	<i>Intuitive Modelle in der Objektorientierten Programmierung</i>	117

11.2	<i>Klassenbegriff</i>	117
11.3	<i>Klasse als Bauplan</i>	117
11.4	<i>Klasse als Fabrik für Objekte</i>	118
11.5	<i>Klasse als Menge von Objekten</i>	118
11.6	<i>Klasse als Prototyp</i>	118
11.7	<i>Klasse als Behälter für Funktionen (Toolbox)</i>	119
<b>12</b>	<b>Objekte</b>	<b>121</b>
12.1	<i>Zustand eines Objektes – Datenbesitz oder holistische Befindlichkeit</i>	121
12.2	<i>Instanzierung von Objekten – Produktion oder Auswahl</i>	122
12.3	<i>Instanzierungsmodelle in der PVS</i>	122
12.4	<i>Interaktion von Objekten – Verarbeitung von Botschaften</i>	124
12.4.1	Objekte als Verursacher von Aktivität	124
12.5	<i>Übermittlung von Botschaften</i>	126
12.6	<i>Schlussfolgerungen</i>	127
<b>13</b>	<b>Intuitive Modellierung</b>	<b>129</b>
13.1	<i>Identifizierung von Entitäten</i>	129
13.2	<i>Abstrahieren</i>	130
13.3	<i>Gestaltungsbildung</i>	131
13.4	<i>Animieren</i>	131
13.5	<i>Clusterbildung und Fokussierung</i>	132
13.6	<i>Überstrukturierung</i>	134
13.7	<i>Einbeziehung der Umgebung</i>	134
13.8	<i>Dekorieren und Dramatisieren</i>	135
13.9	<i>Rückmodellierung</i>	135
13.10	<i>Exhaurierung</i>	136
13.11	<i>Konsistenzwahrung</i>	137
<b>14</b>	<b>Von der Intuition zum Programm</b>	<b>139</b>
14.1	<i>Prozedurale Intuition und fehlendes deklaratives Wissen</i>	139
14.2	<i>Aufbrechen der Gestalt</i>	139
14.3	<i>Fehlende Verbindungen zwischen intuitiven Vorstellungen und Programmkonstrukten</i>	140
14.4	<i>Schwierige intuitive Modelle</i>	140
14.5	<i>Fehlvorstellungen</i>	141
<b>15</b>	<b>Pädagogische Implikationen</b>	<b>143</b>
15.1	<i>Intuitive Modelle und Lernen formaler Programmierkonzepte</i>	143
15.1.1	Molekulare Modelle	143
15.1.2	Atomare Modelle	143
15.1.3	Subatomare Modelle	144
15.2	<i>Kompetenzen im Umgang mit intuitiven Modellen</i>	144
15.2.1	Abstraktionsgrad erkennen	145
15.2.2	Fokus und Grenzen wahrnehmen	145
15.2.3	Medien- und Kommunikationskompetenz	145

<i>15.3 Intuitive Modelle und Scaffolding</i>	<i>146</i>
<i>15.4 Diskussion und Reflektion intuitiver Modelle</i>	<i>146</i>
15.4.1 Visualisierungsübungen	146
15.4.2 Rollenspiele und Regelspiele	147
15.4.3 Gestaltete Medien und Mikrowelten	147
<i>15.5 Informatik im Kontext</i>	<i>148</i>
<i>15.6 Schluss und Ausblick</i>	<i>149</i>
<b>Literatur</b>	<b>151</b>
<b>Anhang</b>	<b>161</b>
<b>Abbildungsverzeichnis</b>	<b>163</b>
<b>Tabellenverzeichnis</b>	<b>167</b>
<b>Abkürzungsverzeichnis</b>	<b>169</b>
<b>1 Ergänzungen zur Repräsentation intuitiver Modelle</b>	<b>171</b>
1.1 Verwendung unterschiedlicher Metaphern beim mathematischen Problemlösen	171
1.2 Repräsentation von Sprachkonzepten durch eine Beispielkollektion – die Python-Kurzreferenz von O'Reilly	171
1.3 Von der Schwierigkeit intuitive Modelle zu visualisieren	172
1.4 Beispiele für Tropen in der Informatik	173
1.5 Mikrowelten als einheitliche Domänen für konzeptionelle Metaphern	173
1.6 Prototypische Beispiele	176
1.7 Beispiele für ablauforientierte Repräsentationen	177
1.8 Darstellung intuitiver Modelle in der Python Visual Sandbox	178
1.8.1 Warum Animationen?	178
1.8.2 Entfernung und Nähe	178
1.8.3 Grafische Elemente der Python Visual Sandbox	179
1.9 Verwendung von Visualisierungen im Informatikunterricht	182
<b>2 Ergänzungen zur Verwendung intuitiver Modelle</b>	<b>185</b>
2.1 Verstehen	185
2.1.1 Textformen in informatischer Fachliteratur	185
2.1.2 Experimente zur Beantwortung erkenntnisgewinnender Fragen	186
2.2 Das Bemühen um Verstehen bei der Vorbereitung auf einen Test	187
2.2.1 Auswahl repräsentativer Beispiele	188
2.2.2 Beispiele ausprobieren – Streben nach Gewissheit	189
2.2.3 Beispiele für die Verwendung regulärer Ausdrücke	189
2.2.4 Ergebnisse	191
2.2.5 Verwendung von visuellen Modellen als Verstehenshilfe	192
2.3 Problemlösen	194
2.3.1 Fallstudie: Das Iterator-Pattern und seine Implementierung in Python	194
2.4 Kontrolle	196
2.4.1 Intuitive Modelle und Testen	196
2.4.2 Paradigmatische Modelle beim Testen	197
<b>3 Materialien zu den empirischen Untersuchungen</b>	<b>199</b>
3.1 Visualisierungsübungen	199
3.1.1 Aufgabenblatt 1	199
3.1.2 Aufgabenblatt 2	200
3.2 Aufbau der Datenbank der PVS	203

3.2.1	Allgemeine Tabellen	203
3.2.2	Tabellen für Python Visual	205
3.2.3	Tabellen für Python Puzzle	206
3.2.4	Tabellen für Python Quiz	208
<b>3.3</b>	<i>Gruppen einrichten</i>	<b>209</b>
<b>3.4</b>	<i>Auswertung der Datenbank</i>	<b>209</b>
3.4.1	Auswertungsmöglichkeiten für Spieler und Zuschauer	210
3.4.2	Wissenschaftliche Auswertung	211
<b>3.5</b>	<i>Python Visual</i>	<b>212</b>
3.5.1	Dokumentation einer Session	212
3.5.2	Auszug aus der automatisch erstellten Auswertung	212
<b>3.6</b>	<i>Python Puzzle</i>	<b>236</b>
3.6.1	Screenshots aus einer Sitzung mit dem Python Puzzle „Modeling a group“	236
3.6.2	Dokumentation einer Session mit XML	236
3.6.3	Beispiel für eine automatisch generierte statistische Auswertung	238
<b>3.7</b>	<i>Python Quiz</i>	<b>240</b>
3.7.1	Dokumentation einer Session	240
3.7.2	Auszug aus der automatisch erstellten Auswertung	241
<b>4</b>	<b>Ergänzungen zu Steuerungsmodellen</b>	<b>309</b>
<b>4.1</b>	<i>Verzweigungen in Datenfluss-Modellen</i>	<b>309</b>
<b>4.2</b>	<i>Der Fetch-Execute-Zyklus als Beispiel einer Schleife</i>	<b>309</b>
<b>4.3</b>	<i>Assozierte Konzepte zur Rekursion</i>	<b>311</b>
<b>4.4</b>	<i>Fehlerhafte Verwendung des Modells der Selbstaufforderung bei eingebetteter Rekursion</i>	<b>312</b>
<b>4.5</b>	<i>Anwendung des Delegationsmodells zur Visualisierung der Arbeitsweise rekursiver Funktionen</i>	<b>312</b>
<b>4.6</b>	<i>Bevorzugung vollständiger Modelle zur Darstellung einer rekursiven Funktion</i>	<b>313</b>
<b>5</b>	<b>Ergänzungen zu Verarbeitungsmodellen</b>	<b>314</b>
<b>5.1</b>	<i>Entstehungsprozesse im Alltag</i>	<b>314</b>
<b>5.2</b>	<i>Vernichtungskonzepte im Alltag</i>	<b>314</b>
<b>5.3</b>	<i>Totale Vernichtung bei Zuweisungen</i>	<b>315</b>
<b>5.4</b>	<i>Beispiele für Datenumwandlungen</i>	<b>316</b>
<b>5.5</b>	<i>Umbenennungen bei der der Ausführung von Funktionen</i>	<b>318</b>
<b>5.6</b>	<i>Umbenennungen in Rechenprotokollen zur rekursiven Berechnung der Fakultät</i>	<b>319</b>
<b>5.7</b>	<i>Datenbewegung bei Iterationen</i>	<b>321</b>
<b>5.8</b>	<i>Namenbewegung bei Iterationen</i>	<b>322</b>
<b>6</b>	<b>Ergänzungen zu intuitiven Modellen in der OOP</b>	<b>323</b>
<b>6.1</b>	<i>Klasse und Schema</i>	<b>323</b>
<b>6.2</b>	<i>Visualisierung von Klassen in Schülerzeichnungen</i>	<b>323</b>
<b>6.3</b>	<i>Klasse als Entität</i>	<b>324</b>
<b>6.4</b>	<i>Prototyptheorien in der Kognitionspsychologie</i>	<b>324</b>
<b>6.5</b>	<i>Prototyporientierte Programmiersprachen</i>	<b>325</b>
<b>6.6</b>	<i>Implizite Verwendung des Prototypkonzepts bei der Entwicklung einer Klasse</i>	<b>325</b>
<b>6.7</b>	<i>Das Prototyp-Konzept bei der Nutzung von Grafik-Tools</i>	<b>326</b>
<b>6.8</b>	<i>Modelle für die Herstellung von Objekten</i>	<b>327</b>

<i>6.9 Modellierung verschachtelter Botschaften</i>	327
6.9.1 Kontexte für die Verwendung von passiven Objektmodellen	329
<i>6.10 Indikatoren für die Validität der Ergebnisse</i>	330
<b>7 Weitere Aspekte der intuitiven Modellierung</b>	<b>330</b>
7.1 <i>Intuitivität als messbare Größe</i>	330
7.2 <i>Überstülpen des EVA-Modells als Beispiel für Überstrukturierung</i>	331