

# INHALT

	VORWORT	VII
	EINFÜHRUNG	IX
<b>1</b>	<b>C++</b>	<b>1</b>
	<i>Bjarne Stroustrup</i>	
	Designentscheidungen	2
	Anwenden der Sprache	6
	OOP und Nebenläufigkeit	10
	Zukunft	14
	Lehren	18
<b>2</b>	<b>PYTHON</b>	<b>21</b>
	<i>Guido von Rossum</i>	
	Der pythonische Weg	22
	Der gute Programmierer	29
	Viele Pythons	35
	Hilfen und Erfahrungen	39
<b>3</b>	<b>APL</b>	<b>45</b>
	<i>Adin D. Falkoff</i>	
	Papier und Stift	46
	Grundlegende Prinzipien	49
	Parallelität	55
	Vermächtnis	58
<b>4</b>	<b>FORTH</b>	<b>61</b>
	<i>Charles H. Moore</i>	
	Die Sprache Forth und das Sprachdesign	62
	Hardware	70
	Anwendungsdesign	73
<b>5</b>	<b>BASIC</b>	<b>81</b>
	<i>Thomas E. Kurtz</i>	
	Die Ziele von BASIC	82
	Compilerdesign	89
	Sprach- und Programmierpraktiken	92
	Sprachdesign	94
	Arbeitsziele	99

<b>6</b>	<b>AWK</b>	<b>103</b>
	<i>Alfred Aho, Peter Weinberger und Brian Kernighan</i>	
	Das Leben von Algorithmen	104
	Sprachdesign	106
	Unix und seine Kultur	108
	Die Rolle der Dokumentation	113
	Informatik	117
	Aufzucht kleiner Sprachen	119
	Entwerfen einer neuen Sprache	124
	Legacy-Kultur	131
	Transformative Technologien	134
	Bits, die das Universum ändern	139
	Theorie und Praxis	145
	Warten auf den Durchbruch	152
	Programming by Example	157
<b>7</b>	<b>LUA</b>	<b>163</b>
	<i>Luiz Henrique de Figueiredo und Roberto Ierusalimsky</i>	
	Die Macht der Skripten	164
	Erfahrung	167
	Sprachdesign	172
<b>8</b>	<b>HASKELL</b>	<b>181</b>
	<i>Simon Peyton Jones, Paul Hudak, Philip Wadler und John Hughes</i>	
	Ein funktionales Team	182
	Trajektorien der funktionalen Programmierung	184
	Die Sprache Haskell	191
	(Funktionales) Wissen verbreiten	198
	Formalisten und Evolution	200
<b>9</b>	<b>ML</b>	<b>207</b>
	<i>Robin Milner</i>	
	Die Solidität von Theoremen	208
	Die Theorie der Bedeutung	216
	Über Informatik hinaus	222
<b>10</b>	<b>SQL</b>	<b>229</b>
	<i>Don Chamberlin</i>	
	Ein bahnbrechender Artikel	230
	Die Sprache	233
	Feedback und Weiterentwicklung	237
	XQuery und XML	242

<b>11</b>	<b>OBJECTIVE-C</b>	<b>247</b>
	<i>Brad Cox und Tom Love</i>	
	Die Entwicklung von Objective-C	248
	Das Wachsen einer Sprache	250
	Ausbildung und Training	255
	Projektmanagement und alte Software	257
	Objective-C und andere Sprachen	264
	Komponenten, Sand und Steine	269
	Qualität als ökonomisches Phänomen	276
	Ausbildung	278
<b>12</b>	<b>JAVA</b>	<b>283</b>
	<i>James Gosling</i>	
	Stärke oder Einfachheit	284
	Eine Frage des Geschmacks	287
	Nebenläufigkeit	290
	Entwerfen einer Sprache	292
	Feedbackschleife	297
<b>13</b>	<b>C#</b>	<b>301</b>
	<i>Anders Hejlsberg</i>	
	Sprache und Design	302
	Wachsen einer Sprache	308
	C#	312
	Die Zukunft der Informatik	317
<b>14</b>	<b>UML</b>	<b>323</b>
	<i>Ivar Jacobson, James Rumbaugh und Grady Booch</i>	
	Lernen und Lehren	324
	Die Rolle der Leute	329
	UML	333
	Wissen	337
	Bereit für Änderungen	340
	Die Verwendung von UML	345
	Schichten und Sprachen	350
	Ein bisschen Wiederverwendbarkeit	354
	Symmetrische Relationen	359
	UML	362
	Sprachdesign	365
	Entwickler ausbilden	371
	Kreativität, Verbesserung und Muster	373

<b>15</b>	<b>PERL</b>	<b>381</b>
	<i>Larry Wall</i>	
	Die Sprache von Revolutionen	382
	Sprache	386
	Community	392
	Evolution und Revolution	396
<b>16</b>	<b>POSTSCRIPT</b>	<b>401</b>
	<i>Charles Geschke und John Warnock</i>	
	Entworfen für die Ewigkeit	402
	Forschung und Bildung	412
	Schnittstellen zur Langlebigkeit	416
	Standardwünsche	420
<b>17</b>	<b>EIFFEL</b>	<b>423</b>
	<i>Bertrand Meyer</i>	
	Ein inspirierender Nachmittag	424
	Wiederverwendbarkeit und Generik	431
	Korrigieren von Sprachen	435
	Wachstum und Evolution	442
	<b>NACHWORT</b>	<b>447</b>
	<b>INTERVIEWPARTNER</b>	<b>449</b>
	<b>INDEX</b>	<b>465</b>