

25	Vererbung	279
25.1	Oberbegriffe und Oberklassen	279
25.2	Ein einfaches Beispiel	280
25.3	Überladen, Überschreiben und Polymorphie	281
25.4	Vererbung in Python.....	284
25.5	Klassenmethoden.....	287
25.6	Standardklassen als Basisklassen	289
26	Mehrfachvererbung	291
26.1	Einführung.....	291
26.2	Beispiel: KalenderUndUhr.....	292
26.3	Diamand-Problem oder „deadly diamond of death“	300
26.4	super und MRO	302
26.4.1	Ein einfaches Beispiel.....	302
26.4.2	Ein umfangreicheres Beispiel.....	305
27	Slots	307
27.1	Erzeugung von dynamischen Attributen verhindern	307
28	Dynamische Erzeugung von Klassen	309
28.1	Beziehung zwischen „class“ und „type“	309
29	Metaklassen	313
29.1	Motivation	313
29.2	Definition	318
29.3	Definition von Metaklassen in Python	318
29.4	Singletons mit Metaklassen erstellen	321
29.5	Beispiel: Methodenaufrufe zählen.....	322
29.5.1	Einführung	322
29.5.2	Vorbereitungen	322
29.5.3	Ein Dekorateur, um Funktionsaufrufe zu zählen	323
29.5.4	Die Metaklasse „Aufrufzähler“	324
30	Abstrakte Klassen	327
31	Aufgaben zur Objektorientierung	331

Teil IV: Funktionale Programmierung	335
32 Begriffsbestimmung	337
33 lambda, map, filter und reduce	339
33.1 lambda	339
33.2 map	341
33.3 Filtern von sequentiellen Datentypen mittels „filter“	344
33.4 reduce	344
33.5 Aufgaben	346
34 Listen-Abstraktion/List Comprehension	347
34.1 Die Alternative zu Lambda und Co.	347
34.2 Syntax	348
34.3 Weitere Beispiele	348
34.4 Die zugrunde liegende Idee	349
34.5 Anspruchsvolleres Beispiel	349
34.6 Mengen-Abstraktion	350
34.7 Rekursive Primzahlberechnung	351
34.8 Generatoren-Abstraktion	351
34.9 Aufgaben	352
35 Generatoren und Iteratoren	353
35.1 Einführung	353
35.2 Iteration in for-Schleifen	353
35.3 Generatoren	355
35.4 Endlos-Generatoren zähmen mit firstn und islice	359
35.5 Sinnvollere Beispiele	360
35.6 Beispiele aus der Kombinatorik	361
35.6.1 Permutationen	361
35.6.2 Variationen und Kombinationen	362
35.7 Generator-Ausdrücke	364
35.8 return-Anweisungen in Generatoren	365
35.9 send-Methode	366
35.10 Die close-Methode	370
35.11 Die throw-Methode	371
35.12 Dekoration von Generatoren	375
35.13 yield from	376
35.14 Aufgaben	378

36	Dekorateur	381
36.1	Einführung Dekorateur	381
36.1.1	Verschachtelte Funktionen	382
36.1.2	Funktionen als Parameter	384
36.1.3	Funktionen als Rückgabewert	385
36.1.4	Fabrikfunktionen	386
36.2	Ein einfacher Dekorateur	388
36.3	@-Syntax für Dekorateur	389
36.4	Anwendungsfälle für Dekorateur	392
36.4.1	Überprüfung von Argumenten durch Dekorateur	392
36.4.2	Funktionsaufrufe mit einem Dekorateur zählen	393
36.5	Dekorateur mit Parametern	395
36.6	Benutzung von Wraps aus functools	396
36.7	Eine Klasse als Dekorateur benutzen	398
36.8	Memoisation	399
36.8.1	Bedeutung und Herkunft des Begriffs	399
36.8.2	Memoisation mit Dekorateurfunktionen	399
36.8.3	Memoisation mit einer Klasse	400
36.8.4	Memoisation mit functools.lru_cache	401
Teil V: Weiterführende Themen		405
37	Tests und Fehler	407
37.1	Einführung	407
37.2	Modultests	409
37.3	Modultests unter Benutzung von <code>__name__</code>	410
37.4	doctest-Modul	412
37.5	Testgetriebene Entwicklung oder „Im Anfang war der Test“	415
37.6	unittest	417
37.7	Methoden der Klasse TestCase	419
37.8	Aufgaben	420
38	Daten konservieren	423
38.1	Persistente Speicherung	423
38.2	Pickle-Modul	424
38.2.1	Daten „einpökeln“ mit <code>pickle.dump</code>	424
38.2.2	<code>pickle.load</code>	425
38.3	Ein persistentes Dictionary mit <code>shelve</code>	425

39	Reguläre Ausdrücke	429
39.1	Ursprünge und Verbreitung	429
39.2	Stringvergleiche	429
39.3	Überlappungen und Teilstrings	431
39.4	Das re-Modul	431
39.5	Matching-Problem	432
39.6	Syntax der regulären Ausdrücke	434
39.6.1	Beliebiges Zeichen	434
39.7	Zeichenauswahl	435
39.8	Endliche Automaten	436
39.9	Anfang und Ende eines Strings	436
39.10	Vordefinierte Zeichenklassen	438
39.11	Optionale Teile	440
39.12	Quantoren	441
39.13	Gruppierungen und Rückwärtsreferenzen	443
39.13.1	Match-Objekte	443
39.14	Iteration über Matches mit finditer	446
39.15	Umfangreiche Übung	446
39.16	Alles finden mit findall	448
39.17	Alternativen	449
39.18	Compilierung von regulären Ausdrücken	450
39.19	Aufspalten eines Strings mit oder ohne regulären Ausdruck	450
39.19.1	split-Methode der String-Klasse	450
39.19.2	split-Methode des re-Moduls	452
39.19.3	Wörter filtern	454
39.20	Suchen und Ersetzen mit sub	455
39.21	Aufgaben	455
40	Typanmerkungen	459
40.1	Einführung	459
40.2	Einfaches Beispiel	460
40.3	Variablenanmerkungen	461
40.4	Listenbeispiele	462
40.5	Listen mit homogenem Typ	463
40.5.1	Version 3.6 bis 3.8	463
40.5.2	Python 3.9 und später	464

41	Systemprogrammierung	467
41.1	Einleitung	467
41.2	Häufig falsch verstanden: Shell	467
41.3	os-Modul	468
41.3.1	Vorbemerkungen	468
41.3.2	Umgebungsvariablen	469
41.3.3	Dateiverarbeitung auf niedrigerer Ebene	471
41.3.4	Weitere Funktionen im Überblick	476
41.3.5	os.path – Arbeiten mit Pfaden	490
41.4	shutil-Modul	498
41.5	glob-Modul	503
42	Forks	505
42.1	Fork	505
42.2	Fork in Python	505
Teil VI: Lösungen zu den Aufgaben		509
43	Lösungen zu den Aufgaben	511
43.1	Lösungen zu Kapitel 5 (Sequentielle Datentypen)	511
43.2	Lösungen zu Kapitel 6 (Listen und Tupel im Detail)	514
43.3	Lösungen zu Kapitel 7 (Verzweigungen)	516
43.4	Lösungen zu Kapitel 8 (Schleifen)	518
43.5	Lösungen zu Kapitel 9 (Dictionaries)	521
43.6	Lösungen zu Kapitel 12 (Dateien lesen und schreiben)	523
43.7	Lösungen zu Kapitel 15 (Funktionen)	524
43.8	Lösungen zu Kapitel 17 (Rekursive Funktionen)	529
43.9	Lösungen zu Kapitel 18 (Sortieren)	534
43.10	Lösungen zu Kapitel 20 (Alles über Strings ...)	537
43.11	Lösungen zu den Kapiteln 22 bis 31 (Aufgaben zur Objektorientierung)	540
43.12	Lösungen zu Kapitel 33 (lambda, map, filter und reduce)	554
43.13	Lösungen zu Kapitel 34 (Listen-Abstraktion/List Comprehension)	555
43.14	Lösungen zu Kapitel 35 (Generatoren und Iteratoren)	556
43.15	Lösungen zu Kapitel 37 (Tests und Fehler)	560
43.16	Lösungen zu Kapitel 39 (Reguläre Ausdrücke)	560
Stichwortverzeichnis		567