

knihovna programátora

- Může sloužit jako učebnice i referenční příručka
- Probírá všechny konstrukce jazyka včetně těch běžně přeskakovaných a některých teprve chystaných
- Všechny probírané konstrukce demonstruje na příkladech
- Vysvětluje principy, na nichž je jazyk postaven, a jejichž znalost umožňuje lépe využívat jeho možnosti
- Pro výklad složitějších konstrukcí používá syntaktické diagramy



RUDOLF PECINOVSKÝ

Python

KOMPLETNÍ PŘÍRUČKA JAZYKA
PRO VERZI 3.10



knihovna programátora

RUDOLF PECINOVSKÝ

Python

**Kompletní příručka
jazyka pro verzi 3.10**

GRADA
Publishing

Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele.

Neoprávněné užití této knihy bude **trestně stíháno**.

Rudolf Pecinovský

Python

Kompletní příručka jazyka pro verzi 3.10

Vydala Grada Publishing, a.s.
U Průhonu 22, Praha 7
obchod@grada.cz, www.grada.cz
tel.: +420 234 264 401
jako svou 8162. publikaci

Odpovědný redaktor Petr Somogyi
Fotografie na obálce Depositphotos/mario7
Grafická úprava a sazba Rudolf Pecinovský
Počet stran 592
První vydání, Praha 2021
Vytiskla tiskárna PBTisk a.s., Příbram

© Grada Publishing, a.s., 2021
Cover Design © Grada Publishing, a. s., 2021
Cover Photo © Depositphotos/mario7

Názvy produktů, firem apod. použité v knize mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

ISBN 978–80–271–4424–2 (ePub)
ISBN 978–80–271–4423–5 (pdf)
ISBN 978–80–271–3442–7 (print)

Všem, kteří se chtějí něco naučit

Stručný obsah

Úvod	25
Část A Superzáklady	33
1 Startujeme	34
2 Zadávání jednoduchých hodnot	51
3 Zadávání textů – stringů	58
4 Volání funkcí	67
5 Jednoduché výrazy	81
6 Proměnné	91
7 Logické hodnoty a operace	109
8 Jednoduché příkazy	120
Část B Složené příkazy	129
9 Moduly	130
10 Vytvoření vlastního modulu	144
11 Definice funkcí	162
12 Parametry, argumenty a lokální proměnné funkcí	174
13 Pokročilé rysy funkcí	187
14 Rozhodování	203
15 Opakování	211
16 Ošetřování chyb	227
Část C Kontejnery	243
17 Seznamy	244
18 N-tice	261
19 Množiny	275
20 Slovníky	286
21 Rozšíření definic funkcí	298
22 Formátování stringů	312
23 Operace s kontejnery	334
24 Práce se soubory	344

Část D Objektově orientované programování	361
25 Základy OOP	362
26 Třídy a jejich instance	374
27 Jednoduché dědění	397
28 Násobné dědění	408
29 Vlastnosti, abstraktní třídy a kachní typování	421
30 Další objektové konstrukce	437
31 Balíčky	451
32 Tvorba aplikací	467
Část E Pokročilejší objektové konstrukce	477
33 Iterátory a generátory	478
34 Přetěžování operátorů	494
35 Anotace a přezdívky datových typů	510
36 Dekorátory	520
37 Ovlivnění přístupu k atributům	531
38 Ovlivnění tvorby tříd, metatřídy	552
39 Korutiny, vlákna, procesy	567
Literatura	576
Rejstřík	578
Část F Přílohy	593
A Konfigurace ve Windows	594
B Syntaktické diagramy	597
C Konvence pro psaní programů v Pythonu	599
D Stručná historie posledních verzí	602
Část G Seznamy	604
Seznam výpisů programů	605
Seznam obrázků	613
Seznam tabulek	614
Seznam odboček – podšeděných bloků	615

Podrobný obsah

Úvod	25
Komu je kniha určena	25
Struktura příručky	27
Koncepte výkladu	27
Jazyk identifikátorů	28
Potřebné vybavení	29
Operační systém	29
Doprovodné programy	29
Použité typografické konvence	30
Odbočka – podšeděný blok	32
Zpětná vazba	32
Část A Superzáklady	33
1 Startujeme	34
1.1 Hlavní součást instalace	34
Platforma	34
Dokumentace	35
PEP	36
Pracovní režimy	36
1.2 Vývojová prostředí	37
PyCharm a IntelliJ IDEA	37
Visual Studio Code	37
Jupyter Notebook a JupyterLab	37
Základní interpret a IDLE	38
1.3 Komunikace s interpretem	38
Odsazování	39
1.4 IDLE – seznamte se	40
Spuštění	40
Základní popis	41
Příkazové okno	42
Restart interaktivního systému	43
Návrat k dříve zadaným příkazům	43
Uložení záznamu seance	44
Editační okno	44
Umístění editovaných souborů	44
Barevné zvýraznění textu	45
1.5 Používání vývojových prostředí	45
Odchyly zobrazení konverzace v IDLE	45
Přepnutí způsobu zobrazení	46
Použité písmo	46
1.6 Objekty a objektové programování	47
Explicitně	47
Objekt, třída, instance, kontejner	47

Objekt.....	48
Třída.....	48
Instance.....	48
Kontejnery.....	48
1.7 Nejdůležitější zvláštnosti Pythonu.....	49
Přísné a benevolentní programovací jazyky.....	49
2 Zadávání jednoduchých hodnot.....	51
2.1 Zápis celých a desetinných (reálných) čísel.....	51
Zpřehlednění dlouhých čísel pomocí znaku podtržení.....	52
2.2 Komplexní čísla.....	53
2.3 Počáteční nula.....	54
2.4 Zadávání čísel v jiných číselných soustavách.....	54
2.5 Platí – neplatí.....	55
2.6 Nic – None.....	55
2.7 Výpustka – Ellipsis,	55
2.8 Objekt NotImplemented.....	56
2.9 Literály.....	56
2.10 Důležitost přehlednosti.....	56
3 Zadávání textů – stringů.....	58
3.1 Zadávání textů.....	58
Escape sekvence.....	60
Bílé znaky.....	62
3.2 Komentáře.....	62
3.3 Slučování sousedních textových literálů.....	63
3.4 Zadání na více řádcích.....	64
3.5 Prefixy stringových literálů.....	64
3.6 Stringová interpolace – f-stringy.....	65
Samodokumentující se výrazy.....	65
Shrnutí zásad pro práci s f-stringy.....	66
4 Volání funkcí.....	67
4.1 Volání funkčního objektu.....	67
Parametr versus argument.....	68
Syntaxe volání a návratová hodnota.....	68
Pořadí vyhodnocování a předávání argumentů.....	69
Datové × funkční objekty.....	69
4.2 Vestavěné funkce s jednoduchými argumenty.....	70
Zápis syntaxe.....	70
Přehled vestavěných funkcí s argumenty jednoduchých typů.....	70
abs(x, /).....	70
ascii(objekt, /).....	71
bin(number, /).....	71
bool(x=False, /).....	71
complex(real=0, imag=0).....	71
divmod(x, y, /).....	71
eval(object, /).....	71
exec(object, /).....	71
float(x=0, /).....	72
hash(obj, /).....	72
help() help(object, /).....	72
hex(number, /).....	72
chr(i, /).....	72
id(object, /).....	72
input(prompt=None, /).....	73

int(x=0, base=10, /)	73
len(obj, /)	73
max(arg1, arg2 ...) min(arg1, arg2 ...)	74
oct(number, /)	74
ord(c, /)	74
pow(base, exp, mod=None)	74
print(argumenty)	74
range(stop) range(start, stop, step=1, /)	75
repr(obj, /)	75
round(number, ndigits=None)	75
str(object='')	76
type(object)	76
4.3 Získání nápovědy	77
Argument zadán	77
Nápověda k některým operátorům a konstrukcím jazyka	78
Bez argumentu	78
4.4 Rozdělení volání na více řádků	79
5 Jednoduché výrazy	81
5.1 Trocha teorie	81
Operace	81
Operand	81
Operátor	82
Arita operátorů	82
Priorita operátorů	83
Asociativita binárních operátorů	83
Operátory jako funkční objekty	85
5.2 Numerické operace	85
Tři druhy dělení	85
Umocňování	86
Nekonečna a nesmyslná čísla	87
5.3 Operace s texty	88
Sečítání textů	88
Násobení textů	89
Indexace jednotlivých znaků	89
6 Proměnné	91
6.1 Co jsou to proměnné	91
Proměnná versus atribut	92
6.2 Správa paměti	92
Statické a dynamické typování	93
6.3 Pravidla pro tvorbu identifikátorů	94
Klíčová slova tvrdá a měkká	94
Systémové identifikátory – dunder	95
6.4 Zavedení proměnné – přiřazovací příkaz	95
Zadání více příkazů na řádku – oddělující středník	97
Zadání skupiny hodnot	97
Proměnné inf a nan	99
6.5 Vnořená volání funkčních objektů	99
6.6 Zjištění typu objektu v proměnné	101
6.7 Uložení funkce do proměnné	102
6.8 Lambda-výrazy	103
Vnořování volání funkčních objektů	103
6.9 Datové a funkční proměnné	104
6.10 Mezery ve výrazech a příkazech	105
6.11 Uložení do proměnné x propojení s názvem	105

6.12	Přřazovací výraz	105
6.13	Pomocné proměnné	107
6.14	F-stringy – rozšřující informace	107
	Další pravidla	107
	Formátování f-stringů	108
7	Logické hodnoty a operace	109
7.1	Konstanty True a False	109
7.2	Převod jiných hodnot na logické	110
7.3	Porovnávání hodnot	111
	Porovnání reálných čísel	111
	Porovnávání a řazení textů – stringů	111
	Zřetěžené porovnávání	112
	Porovnávání totožnosti objektů	113
7.4	Logické operátory a operace	113
	Zkrácené vyhodnocení	114
	Pozor na priority	115
7.5	Operace s jednotlivými bity	115
7.6	Bitové posuny	117
	Aritmetický × logický posun	117
7.7	Podmíněný výraz	118
8	Jednoduché příkazy	120
8.1	Přřkaz pass	120
8.2	Přřkaz tvořený výrazem, výrazový přřkaz	120
8.3	Několik přřkazů na řádku	121
8.4	Přřazovací přřkaz	121
	Složený přřazovací přřkaz	122
8.5	Přřkaz del	122
8.6	Přřkaz assert	123
	Návrh podle kontraktu	124
8.7	Složené přřkazy a odsazování	126
	Výhody a nevýhody koncepce <i>Pythonu</i>	127
	Fyzické a logické řádky	128

Část B Složené přřkazy 129

9	Moduly	130
9.1	Další trocha teorie OOP	130
	Atributy	131
	Práce s objekty – kvalifikace	131
	Vše je součástí nějakého modulu	132
	Dva názvy objektů	132
	Zdrojový soubor	132
	Přeložený soubor	132
9.2	Přřkaz import	133
	Čistý import jiného modulu	133
	Import modulu pod jiným názvem	134
	Přřmý import vyjmenovaných objektů	136
	Import objektů modulu nezahrnuje import jejich modulu	137
	Argumentem přřkazů import a from ... import nesmí být výraz	138
	Import všech atributů daného modulu – hvězdičkový import	138
	Systémové identifikátory	140
	Syntaktické diagramy přřkazu import	140
9.3	Modul jako objekt	141
	Modul builtins a zdánlivě neobjektové programování	142

9.4	Postup systému při importu modulu	142
	Prohledávané složky	143
10	Vytvoření vlastního modulu	144
10.1	Vytvoření vlastního modulu	144
	Kódová stránka	146
	Dokumentační komentář a atribut <code>__doc__</code>	146
	Informace o načítání modulu	147
	Definice datových atributů	147
	Neveřejné atributy	147
	Veřejné atributy	148
	Výrazové příkazy	148
10.2	Kontrolní tisky	148
	Alternativní postup	149
10.3	Průběh importu vytvořeného modulu	149
	Použitelné názvy	151
10.4	Import jako přiřazovací příkaz – shrnutí	152
10.5	Reimport již importovaného modulu	153
	Pozor na přímo importované proměnné	154
	Specifika funkce <code>importlib.reload()</code>	155
	Rozbor chybového hlášení	156
	Důsledky chybného zavedení modulu	156
	Syntaktické chyby	156
10.6	Ještě jednou veřejné atributy	157
10.7	Zprostředkovaný import	159
10.8	Cyklický import	160
11	Definice funkcí	162
11.1	Definice funkce je jen zvláštní přiřazovací příkaz	163
11.2	Definice vlastní funkce	163
	Jednořádková definice	163
	Víceřádková definice	165
	Ukončování definic v interaktivním režimu	165
	Pokračování analýzy kódu	166
	Interaktivní režim versus zdrojový kód modulu	166
	Doporučení	167
	Prázdné funkce	167
11.3	Zadávání stringů zabírajících více řádků	167
11.4	Definice funkcí v modulu	168
11.5	Kdy se projeví chyby v definici funkce	169
11.6	Pomocné funkce pro ladění	171
12	Parametry, argumenty a lokální proměnné funkcí	174
12.1	Parametry, argumenty a lokální proměnné	174
	Definice	174
	Lokální proměnné	175
	Volání funkcí s parametry	176
	Povinně pojmenované argumenty	177
	Povinně poziční argumenty	178
	Mix pozičních a pojmenovaných argumentů	179
12.2	Implicitní hodnoty argumentů	180
12.3	Konstantnost předdefinovaných hodnot	182
12.4	Funkce s vedlejší efektem	183
12.5	Funkce vracějící hodnotu a příkaz <code>return</code>	183
12.6	Přetěžování funkcí	184
	Něco přetížít jde	185
12.7	Anotace	185

13 Pokročilé rysy funkcí	187
13.1 Vnitřní funkce	187
Odsazování	188
13.2 IDLE a nastavení Show Code Context	188
13.3 Jmenné prostory	189
13.4 Oblast/rozsah platnosti, působnost (scope)	190
Zanoření jmenových prostorů	190
13.5 Lokalita použitých proměnných	191
Volná proměnná	191
Příkaz <code>global</code>	193
Příkaz <code>nonLocal</code>	194
13.6 Vnoření funkce versus vnoření volání funkcí	196
13.7 Vnořená volání funkcí	196
13.8 Funkce vyššího řádu	197
13.9 Atributy funkcí	199
13.10 Nelokální proměnné a uzávěry (closures)	199
13.11 Import uvnitř funkce	200
13.12 Možné řešení cyklického importu	201
13.13 Další vlastnosti funkcí	202
14 Rozhodování	203
14.1 Rozhodovací příkazy	203
14.2 Jednoduchý podmíněný příkaz	204
14.3 Úplný podmíněný příkaz	204
14.4 Rozšířený podmíněný příkaz	206
14.5 Přepínač <code>match</code>	207
Trocha terminologie	207
Postup vyhodnocení	208
Sdružování hodnot ve vzorech	209
Klíčové slovo <code>_</code> je jen symbol	209
Další možnosti	210
14.6 Přímé zadání podmíněného příkazu	210
15 Opakování	211
15.1 Rekurze	211
Zásobník návratových adres – ZNA	213
15.2 Příkaz <code>while</code> – cyklus se vstupní podmínkou	214
15.3 Nekonečný cyklus	215
15.4 Příkaz <code>break</code> – cyklus s podmínkou uprostřed	216
15.5 Cyklus s ukončovací podmínkou	217
15.6 Přiřazení v hlavičce cyklu	217
15.7 Větev <code>else</code>	218
15.8 Příkaz <code>continue</code>	219
15.9 Účel a syntaxe cyklu <code>for</code>	219
15.10 Vyjmenování hodnot parametru cyklu	221
15.11 Využití funkce <code>range()</code>	222
Použití indexů	222
15.12 Použití stringu jako zdroje	223
15.13 Vnořování cyklů	224
15.14 Postupné použití několika zdrojů	224
15.15 Větev <code>else</code>	225
16 Ošetřování chyb	227
16.1 Tři druhy chyb	227
Syntaktické chyby	227

Běhové chyby	228
Logické chyby.....	228
16.2 Chybové zprávy	229
Syntaktické chyby při interpretaci příkazu v interaktivním režimu	229
Syntaktické chyby při zadávání příkazu v konzolovém okně	229
Syntaktické chyby při překladu importovaného modulu	230
Běhové chyby	230
16.3 I chyby jsou objekty – výjimky	232
16.4 Rozdělení výjimek	233
16.5 Zachycení a ošetření výjimky	234
16.6 Více větví except.....	235
16.7 Větev else.....	235
16.8 Větev finally	235
16.9 Syntaktický diagram příkazu try	236
16.10 Příklad s kompletní verzí příkazu try.....	237
Převod se nepodařil.....	237
Převod se podařil.....	237
16.11 Praktický příklad	238
16.12 Zdánlivé záludnosti větve finally	239
16.13 Vyhození výjimky.....	242
16.14 Příkaz assert.....	242
16.15 Hierarchie výjimek a definice vlastní výjimky.....	242

Část C Kontejnery

243

17 Seznamy.....	244
17.1 Proměnné a neměnné objekty.....	244
17.2 Tvorba instancí a konstruktory	245
17.3 Základní informace o seznamech	246
17.4 Vytváření seznamů	246
Použití literálu	247
Využití konstrukturu list(seq=())	247
Sčítání a násobení.....	248
17.5 Generátorová notace seznamů	249
17.6 Modifikace seznamů.....	251
Metody append() a extend()	251
Rizika práce s odkazy na proměnné objekty	252
Postupné budování seznamu.....	253
Přičítání jiných zdrojů	254
Indexace prvků seznamu	255
Metody pracující s indexy.....	256
index(value, start=0, stop=9223372036854775807, /) -> int	256
insert(index, object, /)	257
pop(index=-1, /) -> ?	257
remove(self, value, /)	257
Metody pracující s celým seznamem	257
reverse(self, /)	257
sort(*, key=None, reverse=False)	257
17.7 Vícerozměrné seznamy	258
17.8 Souhrnný příklad	258
17.9 Anotace odkazující na seznamy.....	260
18 N-tice	261
18.1 Základní informace o n-ticích	261

18.2	Vytváření n-tic	261
	Vytváření n-tic pomocí literálů	262
	Využití konstruktoru <code>tuple(seq=())</code>	263
	Sčítání a násobení	264
	Přičítání n-tic	265
	Balení a rozbalování n-tic	266
	Prohazování proměnných	266
	Hvězdičkové pravidlo	267
18.3	Generátorová notace n-tic	267
18.4	Problematika neměnnosti n-tic	268
	Hešovatelné objekty	269
18.5	Přístup k prvkům n-tic	269
18.6	Sčítání seznamů a n-tic	270
18.7	Proměnné a neměnné prvky n-tice	270
18.8	Pojmenované n-tice	272
18.9	Anotace odkazující na n-tice	274
19	Množiny	275
19.1	Základní informace o množinách	275
19.2	Vytváření množin	275
	Vytváření množin pomocí literálů	276
	Vytváření množin pomocí konstruktoru <code>set()</code>	276
	Hešová tabulka	276
	Použitelné a nepoužitelné zdroje	277
	Vytváření množin prostřednictvím množinových operací	278
	<code>union(*zdroj) a b ...</code>	279
	<code>intersection(*zdroj) a & b & ...</code>	279
	<code>difference(*zdroj) a - b - ...</code>	279
	<code>symmetric_difference(zdroj) a ^ b</code>	280
19.3	Generátorová notace množin	280
19.4	Zmražené množiny	280
19.5	Modifikace množin	281
	Modifikace pracující s jedním prvkem	281
	<code>add(element)</code>	282
	<code>discard(element)</code>	282
	<code>remove(element)</code>	282
	<code>pop()</code>	282
	Množinové operátory a sdružené operace	283
	<code>update(*zdroj) a = b ...</code>	284
	<code>intersection_update(*zdroj) a &= b & ...</code>	284
	<code>difference_update(*zdroj) a -= b ...</code>	284
	<code>symmetric_difference_update(zdroj) a ^= b</code>	284
	Porovnávání množin	284
	<code>isdisjoint(množina)</code>	284
	<code>a < b</code>	284
	<code>issubset(množina) a <= b</code>	285
	<code>issuperset(množina) a >= b</code>	285
	<code>a > b</code>	285
19.6	Anotace odkazující na množiny	285
20	Slovníky	286
20.1	Mapovací objekty a slovníky	286
20.2	Vytváření slovníků	287
	Vytváření slovníků pomocí literálů	287
	Vytváření slovníků pomocí konstruktoru <code>dict()</code>	288
	Ekvivalence slovníků	289

Vytváření slovníků pomocí metody <code>fromkeys()</code>	289
20.3 Generátorová notace slovníků	290
20.4 Operace se slovníkem	291
Práce s hodnotami pomocí „indexace“ klíčem	291
Další metody pro práci s jednotlivými položkami	293
<code>get(key, default=None, \)</code>	293
<code>pop(key, default=#, \)</code>	293
<code>popitem()</code>	294
<code>setdefault(key, default=None, \)</code>	294
Modifikace slovníku daty ze zadaného zdroje	294
<code>update(zdroj)</code>	294
Operátor <code> </code>	294
Sdružené přiřazení <code> =</code>	294
Slovník jako generátor	295
20.5 Pohledy	295
<code>items()</code>	295
<code>keys()</code>	295
<code>values()</code>	295
Pohledy jako zdroje dat	296
Operace s pohledy	296
20.6 Anotace odkazující na slovníky	297
21 Rozšíření definic funkcí	298
21.1 Předávání argumentů odkazem a hodnotou	298
Předání argumentu odkazem	299
21.2 Pomocná funkce <code>gr()</code>	300
21.3 Proměnný počet pozičních argumentů	301
Hvězdičkový parametr	301
Hvězdičkový argument	303
21.4 Proměnný počet pojmenovaných argumentů	304
Dvuhvězdičkový parametr	304
Dvuhvězdičkový argument	304
21.5 Stručný souhrn	305
Podivné chování	306
Použití v definicích literálů	307
21.6 Vestavěné funkce pracující s kontejnery	307
<code>all(iterable)</code>	307
<code>any(iterable)</code>	307
<code>dir(object)</code>	308
<code>enumerate(iterable, start=0)</code>	308
<code>eval(expression, globals=None, locals=None, /)</code>	308
<code>exec(source, globals=None, locals=None, /)</code>	309
<code>filter(function, iterable)</code>	309
<code>globals()</code>	309
<code>len(c)</code>	309
<code>locals()</code>	309
<code>max(iterable, */, key, default)</code>	310
<code>max(arg1, arg2, *args/, key)</code>	310
<code>min(iterable, */, key, default)</code>	310
<code>min(arg1, arg2, *args/, key)</code>	310
<code>reversed(seq)</code>	310
<code>slice(stop) slice(start, stop/, step)</code>	310
<code>sorted(iterable, *, key=None, reverse=False)</code>	310
<code>sum(iterable, /, start=0)</code>	311
<code>vars(object)</code>	311

zip(*iterables)	311
22 Formátování stringů.....	312
22.1 Formátovací operátor %.....	312
22.2 Pokročilejší metody formátování	314
Metoda format() versus f-string	314
Formátovací string.....	314
Formát nahrazovacího pole	315
Formát nahrazovaného textu.....	316
Konverze	317
22.3 Specifikace formátu.....	318
Počet zabraných pozic	318
Přesnost.....	319
Typ hodnoty	320
Stringy	320
Celočíselné hodnoty	320
Numerické hodnoty.....	321
Skupiny číslic	322
Alternativní formát a vedoucí nuly.....	324
Znaménko	325
Zarovnání a plnění	325
Vnořená nahrazovací pole	327
Formátování samodokumentujících se nahrazovacích polí.....	327
22.4 Příklad: Pascalův trojúhelník.....	328
22.5 Příklad: Trasovací funkce prSE() a prIN()	329
Funkce prSE()	329
Funkce prIN()	331
Použití	332
23 Operace s kontejnery.....	334
23.1 Proměnné objekty jako implicitní hodnoty parametrů	334
23.2 Kopírování	334
Mělké a hluboké kopie objektů	335
Zdánlivé kopie neměnných kontejnerů	336
Alternativní způsob tvorby mělkých kopií.....	336
Nebezpečí hlubokých kopií.....	336
23.3 Rozdělení doposud probraných kontejnerů	337
23.4 Přítomnost prvku v kontejneru.....	338
23.5 Řazení prvků posloupnosti.....	338
reversed(seq)	338
sorted(iterable, *, key=None, reverse=False)	339
23.6 Vykrajování (slicing).....	340
23.7 Indexování a vykrajování u rozsahů	341
23.8 Nahrazování hodnot	342
23.9 „Úprava“ neměnných objektů	343
24 Práce se soubory	344
24.1 Soubory: bleskové opakování.....	344
Soubor, souborový systém, cesta	345
Absolutní a relativní cesta	345
Substituované disky ve Windows	346
Odchylky Pythonu od některých jiných jazyků	346
Dva způsoby práce se souborovým systémem	346
24.2 Moduly os a os.path	348
24.3 Pracovní složka.....	349
24.4 Skládání a rozkládání cest	350
24.5 Vytváření a mazání složek	351

Mazání.....	352
24.6 Získání informací o souborech.....	353
24.7 Zápis a čtení dat.....	354
Otevírání souborů.....	354
Zápis dat, splachování a zavírání souborů.....	356
24.8 Čtení ze souborů.....	357

Část D Objektově orientované programování 361

25 Základy OOP.....	362
25.1 Základní princip OOP.....	363
25.2 Objekty a jejich atributy.....	364
Terminologická vsuvka.....	365
25.3 Třídy a jejich instance.....	366
Třída versus datový typ.....	366
Instance.....	366
25.4 Objekt třídy versus instance třídy.....	367
25.5 Atributy třídy versus atributy instancí.....	367
25.6 Zprávy.....	368
25.7 Metody.....	368
25.8 Dědění.....	369
Terminologie.....	369
LSP – substituční princip Liskové.....	370
Virtuální metody a jejich přebíjení.....	370
Zakrývání versus přebíjení.....	371
Polymorfismus.....	371
Rodičovský podobjekt.....	371
Násobné dědění a diamantový problém.....	371
Zobecňování.....	373
26 Třídy a jejich instance.....	374
26.1 Definice třídy a jejích atributů.....	374
Definice třídy je příkaz.....	376
26.2 Práce s atributy objektu.....	376
Získání a modifikace hodnoty atributu.....	376
Přidání a odebrání atributu.....	377
Nezveřejňované atributy.....	379
Třída jako parametr.....	379
26.3 Instance a práce s nimi.....	379
Textový podpis instance.....	380
Kvalifikace atributu třídy instancí.....	380
Tři druhy metod, použití dekorátorů.....	381
Metody instancí a parametr <code>self</code>	383
Vytváření instancí – konstruktor, alokátor, initor.....	384
Instanční metody třídy C2.....	386
Použití instančních a třídních atributů.....	387
Změny instančních a třídních datových atributů.....	388
Atribut třídy jako výchozí hodnota instančního atributu.....	389
Zavádění nových třídních funkčních atributů.....	390
Zavádění nových instančních funkčních atributů.....	391
26.4 Vestavěné třídy jsou nemodifikovatelné.....	392
26.5 Nutnost kvalifikace atributů.....	392
26.6 Některé speciální atributy – dundery.....	394
<code>__bases__</code>	394
<code>__class__</code>	394
<code>__dict__</code>	394

__doc__	394
__mro__	394
__name__	394
__qualname__	394
__subclasses__()	394
Ukázky použití	395
26.7 Alternativní definice třídy	396
27 Jednoduché dědění	397
27.1 Vytváření potomka a jeho vlastnosti	397
27.2 Příklad	398
Třída LA	398
Třídy LB a LC	400
27.3 Jmenné prostory	400
Atributy a jmenné prostory tříd	400
Atributy a jmenné prostory instancí	401
27.4 Volání metod v hierarchii dědění	403
27.5 Initory se dědí	403
27.6 Hierarchie výjimek	404
27.7 Definice vlastní výjimky	405
27.8 Vestavěné funkce pracující s objekty	406
callable(object)	406
classmethod(method, *args, **kwds)	406
delattr(object, name)	406
getattr(object, name[, default])	406
hasattr(object, name)	407
isinstance(object, classinfo)	407
issubclass(class, classinfo)	407
setattr(object, name, value)	407
staticmethod(method, *args, **kwds)	407
super([type/, object-or-type])	407
28 Násobné dědění	408
28.1 Python a násobné dědění	408
28.2 Příklad: jednoduchý diamant	409
28.3 Analýza chování	410
Přímé zadání volaného initoru	411
Problémy s parametry a argumenty – hubnoucí parametr	413
28.4 Virtuální metody	415
28.5 Složitější hierarchie dědění a MRO	415
Nerealizovatelná a následně opravená hierarchie	415
Zásady specifikace MRO	416
28.6 Komolení jmen a pseudosoukromé atributy	417
28.7 Šablonová metoda	418
29 Vlastnosti, abstraktní třídy a kachní typování	421
29.1 Atributy × vlastnosti	421
29.2 Přímá definice vlastnosti	422
Využití lambda-výrazů, atributy jen pro čtení	424
Vztah vlastnosti k třídě a instancím	424
29.3 Zadání vlastnosti pomocí dekorátoru	426
Použití	427
Shrnutí	429
29.4 Abstraktní × konkrétní třídy a metody	429
Princip abstraktních metod	429
Abstraktní a konkrétní třídy v mainstreamových jazycích	430

Formálně a skutečně abstraktní třídy v jazyce Python	430
29.5 Definice skutečně abstraktní třídy	431
Přebíjet musí i ti, kteří abstraktní metodu nepotřebují	432
Definice potomka abstraktní třídy	432
Proč definovat metodu použitou v šabloně	433
Proč dekorovat metodu jako abstraktní	433
Proč v hlavičce deklarovat předka ABC	433
Shrnutí	433
29.6 Abstraktní vlastnosti a statické a třídí metody	434
29.7 Rozhraní × implementace; protokol	435
Protokol	436
29.8 Kachní typování	436
30 Další objektové konstrukce	437
30.1 Výčtové typy – základy	437
Trocha terminologie	437
Výčtový typ definovaný voláním funkce	438
Vlastnosti výčtového typu a jeho instancí	438
Útroby výčtového typu	440
Výčtový typ definovaný jako třída	440
30.2 Další možnosti výčtových typů	442
Přezdívky hodnot	442
Automatické přiřazení obalovaných hodnot	443
Poloautomatické přiřazení obalovaných hodnot	444
Definice „obyčejných“ atributů	445
Výčtové typy s více předky	445
30.3 Datové třídy	445
30.4 Srovnávání objektů se vzory v příkazu match	447
Opakování terminologie	447
Dosazování hodnot	447
Prověřování posloupností	448
Vzory s výběrem hodnot	449
31 Balíčky	451
31.1 Balíčky (packages)	451
Initor standardního balíčku	451
Implementace	452
Úpravy modulu v balíčku a jeho opětovné načtení	453
Nedostupnost balíčků neimportovaných explicitně	455
31.2 Relativní import	455
31.3 Hromadný import modulů balíčku	456
Příklad	456
Deklarace hromadného importu	459
Opakování: reimport modulu	459
Hvězdičkový import	460
31.4 Neinicializované balíčky – NS-balíčky	461
Teoretický úvod	461
Postup hledání modulu	462
Důsledky	462
Příklad	463
31.5 Standardní balíček v NS-balíčku	465
32 Tvorba aplikací	467
32.1 Vytvoření aplikace či knihovny	467
32.2 Přímé spuštění zadaného skriptu	467
Rozpoznání režimu, v němž byl modul spuštěn	468
Demonstrace	468
32.3 Vytvoření spustitelné aplikace	469