

knihovna programátora

- Učebnice jazyka i referenční příručka
- Vysvětluje principy, na nichž je jazyk postaven a jejichž znalost umožňuje lépe využívat jeho možnosti
- Probírá všechny konstrukce jazyka včetně obvykle opomíjených
- Pro výklad většiny konstrukcí používá syntaktické diagramy
- Všechny probírané konstrukce vysvětluje na příkladech



RUDOLF PECINOVSKÝ

Python

Kompletní příručka jazyka

PRO VERZI 3.11



knihovna programátora

RUDOLF PECINOVSKÝ

Python

**Kompletní příručka
jazyka pro verzi 3.11**

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 3.11

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 600

První vydání, Praha 2023

Tisk: Iva Vodáková – Durabo

© Grada Publishing, a.s., 2023

Cover Design © Grada Publishing, a. s., 2023

Cover Photo © Depositphotos/mario7

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

ISBN 978–80–271–6740-1 (pdf)

ISBN 978–80–271–3891-3 (print)

*Mé ženě Jarušce a dětem
Štěpánce, Pavlínce, Ivance a Michalovi*

Stručný obsah

Úvod	26
Část A Superzáklady	35
1 Startujeme	36
2 Zadávání jednoduchých hodnot	53
3 Zadávání textů – stringů	60
4 Volání funkcí	71
5 Jednoduché výrazy	85
6 Proměnné	95
7 Logické hodnoty a operace	114
8 Jednoduché příkazy	124
Část B Složené příkazy	133
9 Moduly	134
10 Vytvoření vlastního modulu	148
11 Definice funkcí	167
12 Parametry, argumenty a lokální proměnné funkcí	177
13 Pokročilé rysy funkcí	195
14 Rozhodování	212
15 Opakování	220
16 Ošetřování chyb	236
Část C Kontejnery	253
17 Seznamy	254
18 N-tice	271
19 Množiny	284
20 Slovníky	295
21 Rozšíření definic funkcí	307
22 Formátování stringů	321
23 Operace s kontejnery	343
24 Práce se soubory	353

Část D Objektivě orientované programování	373
25 Základy OOP	374
26 Třídy a jejich instance	387
27 Jednoduché dědění.....	411
28 Násobné dědění.....	422
29 Vlastnosti, abstraktní třídy a kachní typování	435
30 Další objektové konstrukce	451
31 Balíčky.....	466
32 Tvorba aplikací	482
Část E Pokročilejší objektové konstrukce	491
33 Iterátory a generátory	492
34 Přetěžování operátorů	508
35 Anotace a přezdívky datových typů	524
36 Dekorátory	534
37 Ovlivnění přístupu k atributům	545
38 Ovlivnění tvorby tříd, metatřídy	566
39 Korutiny, vlákna, procesy.....	581
Literatura	590
Rejstřík	592
Část F Přílohy	599
A Konfigurace ve Windows	600
B Syntaktické diagramy	603
C Konvence pro psaní programů v Pythonu.....	605
D Stručná historie posledních verzí	609
Část G Seznamy	615
Seznam výpisů programů.....	616
Seznam obrázků	624
Seznam tabulek	625
Seznam odboček – podšeděných bloků	626

Podrobný obsah

Úvod	26
Komu je kniha určena	26
Struktura příručky.....	28
Koncepte výkladu	29
Jazyk identifikátorů	29
Potřebné vybavení.....	30
Operační systém.....	30
Doprovodné programy.....	30
Použité typografické konvence	31
Odbočka – podšeděný blok.....	33
Zpětná vazba	33
Část A Superzáklady	35
1 Startujeme	36
1.1 Hlavní součást instalace	36
Platforma	36
Dokumentace	37
PEP	37
Pracovní režimy	38
1.2 Vývojová prostředí	39
PyCharm a IntelliJ IDEA.....	39
Visual Studio Code.....	39
Jupyter Notebook a JupyterLab	40
Základní interpret a IDLE	40
1.3 Komunikace s interpretem.....	41
Odsazování	41
1.4 IDLE – seznamte se	42
Spuštění	42
Základní popis	43
Příkazové okno	44
Restart interaktivního systému	45
Návrat k dříve zadaným příkazům	45
Uložení záznamu seance	45
Editační okno	46
Umístění editovaných souborů.....	46
Barevné zvýraznění textu	47
1.5 Používání vývojových prostředí.....	47
Odchyly zobrazení konverzace v IDLE.....	47
Použité písmo	47
1.6 Objekty a objektové programování.....	48
Explicitně.....	49
Objekt, třída, instance, kontejner.....	49
Objekt	49
Třída	49
Instance	49
Kontejnery	50

1.7	Datový typ	50
1.8	Nejdůležitější zvláštnosti Pythonu	51
	Přísné a benevolentní programovací jazyky	51
2	Zadávání jednoduchých hodnot	53
2.1	Zápis celých a desetinných (reálných) čísel	53
	Zpřehlednění dlouhých čísel pomocí znaku podtržení	54
2.2	Komplexní čísla	55
2.3	Počáteční nula	56
2.4	Zadávání čísel v jiných číselných soustavách	56
2.5	Platí – neplatí	57
2.6	Nic – None	57
2.7	Výpustka – Ellipsis,	58
2.8	Objekt NotImplemented	58
2.9	Literály	58
2.10	Důležitost přehlednosti	59
3	Zadávání textů – stringů	60
3.1	Zadávání textů	60
3.2	Komentáře	62
	Escape sekvence	63
	Bílé znaky	65
3.3	Slučování sousedních textových literálů	65
3.4	Zadání na více řádcích	66
3.5	Prefixy stringových literálů	67
3.6	Stringová interpolace – f-stringy	68
	Samodokumentující se výrazy	68
	Shrnutí zásad pro práci s f-stringy	68
3.7	Bajtové objekty	69
	Bajtové stringy a bajtové literály	69
	Třída bytearray – zadávání bajtových polí	70
	Další informace	70
4	Volání funkcí	71
4.1	Volání funkčního objektu	71
	Parametr versus argument	72
	Syntaxe volání a návratová hodnota	73
	Pořadí vyhodnocování a předávání argumentů	73
	Datové × funkční objekty	73
4.2	Vestavěné funkce s jednoduchými argumenty	74
	Zápis syntaxe	74
	Přehled vestavěných funkcí s argumenty jednoduchých typů	74
	abs(x, /)	75
	ascii(objekt, /)	75
	bin(number, /)	75
	bool(x=False, /)	75
	complex(real=0, imag=0)	75
	divmod(x, y, /)	75
	eval(object, /)	76
	exec(object, /)	76
	float(x=0, /)	76
	hash(obj, /)	76
	help() help(object, /)	76
	hex(number, /)	76
	chr(i, /)	77
	id(object, /)	77

input(prompt=None, /)	77
int(x=0, base=10, /)	77
len(obj, /)	78
list(x)	78
max(arg1, arg2 ...) min(arg1, arg2 ...)	78
oct(number, /)	78
ord(c, /)	78
pow(base, exp, mod=None)	79
print(argumenty)	79
range(stop) range(start, stop, step=1, /)	79
repr(obj, /)	79
round(number, ndigits=None)	79
str(object='')	80
type(object)	81
4.3 Získání nápovědy	81
Argument zadán	81
Nápověda k některým operátorům a konstrukcím jazyka	82
Bez argumentu	82
4.4 Rozdělení volání na více řádků	84
5 Jednoduché výrazy	85
5.1 Trocha teorie	85
Operace	85
Operand	85
Operátor	86
Arita operátorů	86
Priorita operátorů	87
Asociativita binárních operátorů	87
Operátory jako funkční objekty	89
5.2 Numerické operace	89
Tři druhy dělení	89
Umocňování	90
Nekonečna a nesmyslná čísla	91
5.3 Operace s texty	92
Sčítání textů	92
Násobení textů	93
Indexace jednotlivých znaků	93
6 Proměnné	95
6.1 Co jsou to proměnné	95
Proměnná versus atribut – kvalifikace	96
6.2 Správa paměti	96
Statické a dynamické typování	97
6.3 Pravidla pro tvorbu identifikátorů	98
Klíčová slova tvrdá a měkká	98
Systémové identifikátory – dundery	99
6.4 Zavedení proměnné – přiřazovací příkaz	99
Zadání více příkazů na řádku – oddělovací středník	101
Zadání skupiny hodnot	101
Proměnné inf a nan	103
6.5 Vnořená volání funkčních objektů	103
6.6 Zjištění typu objektu v proměnné	104
6.7 Uložení funkce do proměnné	106
6.8 Lambda-výrazy	107
Vnořování volání funkčních objektů	108
Využití při snižování počtu argumentů – currying	108
6.9 Datové a funkční proměnné	109

6.10	Mezery ve výrazech a příkazech.....	109
6.11	Uložení do proměnné × propojení s názvem.....	110
6.12	Přiřazovací výraz	110
6.13	Pomocné proměnné	111
6.14	F-stringy – rozšiřující informace	112
	Další pravidla.....	112
	Formátování f-stringů.....	112
7	Logické hodnoty a operace	114
7.1	Konstanty True a False	114
7.2	Převod jiných hodnot na logické	115
7.3	Porovnávání hodnot.....	116
	Porovnání reálných čísel.....	116
	Porovnávání a řazení textů – stringů	116
	Zřetěžené porovnávání.....	117
	Porovnávání totožnosti objektů	117
7.4	Logické operátory a operace.....	118
	Zkrácené vyhodnocení.....	119
	Pozor na priority	120
7.5	Operace s jednotlivými bity	120
7.6	Bitové posuny	122
	Aritmetický × logický posun	122
7.7	Podmíněný výraz	123
8	Jednoduché příkazy	124
8.1	Příkaz pass.....	124
8.2	Příkaz tvořený výrazem, výrazový příkaz	124
8.3	Několik příkazů na řádku	125
8.4	Přiřazovací příkaz	125
	Složený přiřazovací příkaz.....	126
8.5	Příkaz del	126
8.6	Příkaz assert.....	127
	Návrh podle kontraktu.....	128
8.7	Složené příkazy a odsazování	130
	Výhody a nevýhody koncepce <i>Pythonu</i>	131
	Fyzické a logické řádky	132
	Část B Složené příkazy	133

9	Moduly.....	134
9.1	Další trocha teorie OOP	134
	Atributy	135
	Práce s objekty – kvalifikace	135
	Vše je součástí nějakého modulu	136
	Dva názvy objektů.....	136
	Zdrojový soubor.....	136
	Přeložený soubor.....	136
9.2	Příkaz import.....	137
	Čistý import jiného modulu	137
	Import modulu pod jiným názvem	139
	Přímý import vyjmenovaných objektů.....	140
	Import objektů modulu nezahrnuje import jejich modulu	141
	Argumentem příkazů import a from ... import nesmí být výraz.....	142
	Import všech atributů daného modulu – hvězdičkový import	142
	Systémové identifikátory	144
	Syntaktické diagramy příkazu import.....	144

9.3 Modul jako objekt	145
Modul <code>builtins</code> a zdánlivě neobjektové programování.....	146
9.4 Postup systému při importu modulu	146
Prohledávané složky	147
10 Vytvoření vlastního modulu	148
10.1 Vytvoření vlastního modulu.....	148
Kódová stránka.....	150
Dokumentační komentář a atribut <code>__doc__</code>	150
Informace o načítání modulu	151
Definice datových atributů	151
Neveřejné atributy	151
Veřejné atributy	152
Výrazové příkazy	152
10.2 Kontrolní tisky.....	152
Alternativní postup.....	153
10.3 Průběh importu vytvořeného modulu.....	153
Použitelné názvy.....	155
10.4 Import jako přiřazovací příkaz – shrnutí.....	156
10.5 Reimport již importovaného modulu	157
Pozor na přímo importované proměnné	158
Specifika funkce <code>importlib.reload()</code>	159
Rozbor chybového hlášení.....	160
Důsledky chybného zavedení modulu	160
Syntaktické chyby	160
10.6 Ještě jednou veřejné atributy	161
10.7 Zprostředkovaný import	163
10.8 Cyklický import.....	164
11 Definice funkcí	167
11.1 Definice funkce je jen zvláštní přiřazovací příkaz	168
11.2 Definice vlastní funkce.....	168
Jednořádková definice.....	168
Víceřádková definice.....	170
Ukončování definic v interaktivním režimu	170
Pokračování analýzy kódu.....	171
Interaktivní režim versus zdrojový kód modulu	171
Doporučení	172
Prázdné funkce	172
11.3 Zadávání stringů zabírajících více řádků.....	172
11.4 Definice funkcí v modulu	173
11.5 Kdy se projeví chyby v definici funkce.....	174
12 Parametry, argumenty a lokální proměnné funkcí.....	177
12.1 Parametry, argumenty a lokální proměnné	177
Definice.....	177
Lokální proměnné.....	178
Volání funkcí s parametry.....	179
Povinně pojmenované argumenty	180
Povinně poziční argumenty.....	181
Mix pozičních a pojmenovaných argumentů.....	182
12.2 Implicitní hodnoty argumentů	183
12.3 Konstantnost předdefinovaných hodnot	185
12.4 Funkce s vedlejší efektem	186
12.5 Funkce vracějící hodnotu a příkaz <code>return</code>	186
12.6 Přetěžování funkcí.....	187
Něco přetížít jde.....	188

12.7	Anotace.....	189
12.8	Dekorátory.....	189
12.9	Pomocné funkce a dekorátory pro ladění.....	190
	prSE(level:int, se:bool, caller:str, msg:str='') -> str.....	190
	prIN(level:int, msg:str='') -> None.....	191
	reset() -> None.....	191
	Dekorátory.....	192
	prSEd(level:int=1, print_args=False, print_res=False, msg:str='', wait=False).....	192
	prSEda(level:int=1, msg:str='').....	192
	prSEdar(level:int=1, msg:str='').....	192
	prSEdr(level:int=1, msg:str='').....	192
	prSEdw(...) prSEdaw(...) prSEdarw(...) prSEdrw(...).....	192
	Příklad.....	192
13	Pokročilé rysy funkcí.....	195
13.1	Vnitřní funkce.....	195
	Odsazování.....	196
	Přístup z okolního kódu.....	197
	Import uvnitř funkce.....	197
13.2	IDLE a nastavení Show Code Context.....	197
13.3	Jmenné prostory.....	198
13.4	Oblast/rozsah platnosti, působnost (scope).....	198
	Zanoření jmenových prostorů.....	199
13.5	Lokalita použitých proměnných.....	200
	Volná proměnná.....	200
	Příkaz global.....	201
	Příkaz nonLocal.....	202
13.6	Vnoření funkce versus vnoření volání funkcí.....	205
13.7	Vnořená volání funkcí.....	205
13.8	Funkce vyššího řádu.....	205
13.9	Atributy funkcí.....	207
13.10	Nelokální proměnné a uzávěry (closures).....	208
13.11	Import uvnitř funkce.....	209
13.12	Možné řešení cyklického importu.....	209
13.13	Další vlastnosti funkcí.....	211
14	Rozhodování.....	212
14.1	Rozhodovací příkazy.....	212
14.2	Jednoduchý podmíněný příkaz.....	213
14.3	Úplný podmíněný příkaz.....	213
14.4	Rozšířený podmíněný příkaz.....	215
14.5	Přepínač match.....	216
	Trocha terminologie.....	216
	Postup vyhodnocení.....	217
	Sdružování hodnot ve vzorech.....	218
	Klíčové slovo _ je jen symbol.....	218
	Další možnosti.....	219
14.6	Přímé zadání podmíněného příkazu.....	219
15	Opakování.....	220
15.1	Rekurze.....	220
	Zásobník návratových adres – ZNA.....	222
15.2	Příkaz while – cyklus se vstupní podmínkou.....	223
15.3	Nekonečný cyklus.....	224
15.4	Příkaz break – cyklus s podmínkou uprostřed.....	225

15.5	Cyklus s ukončovací podmínkou	226
15.6	Přiřazení v hlavičce cyklu	226
15.7	Větev else	227
15.8	Příkaz continue	228
15.9	Účel a syntaxe cyklu for	228
15.10	Vyjmenování hodnot parametru cyklu	230
15.11	Využití funkce range ()	231
	Použití indexů	231
15.12	Použití stringu jako zdroje	232
15.13	Vnořování cyklů	233
15.14	Postupné použití několika zdrojů	233
15.15	Větev else	234
16	Ošetřování chyb	236
16.1	Tři druhy chyb	236
	Syntaktické chyby	236
	Běhové chyby	237
	Logické chyby	237
16.2	Chybové zprávy	238
	Syntaktické chyby při interpretaci příkazu v interaktivním režimu	238
	Syntaktické chyby při zadávání příkazu v konzolovém okně	238
	Syntaktické chyby při překladu importovaného modulu	239
	Běhové chyby	239
16.3	I chyby jsou objekty – výjimky	241
16.4	Rozdělení výjimek	242
16.5	Zachycení a ošetření výjimky	243
16.6	Více větví except	244
16.7	Větev else	244
16.8	Větev finally	244
16.9	Syntaktický diagram příkazu try	245
16.10	Příklad s kompletní verzí příkazu try	246
	Převod se nepodařil	246
	Převod se podařil	246
16.11	Praktický příklad	247
16.12	Zdánlivé záludnosti větve finally	248
16.13	Vyhození výjimky	251
16.14	Příkaz assert	251
16.15	Doplnění výjimky o poznámku	252
16.16	Hierarchie výjimek a definice vlastní výjimky	252

Část C Kontejnery

253

17	Seznamy	254
17.1	Proměnné a neměnné objekty	254
17.2	Tvorba instancí a konstruktory	255
17.3	Základní informace o seznamech	256
17.4	Vytváření seznamů	256
	Použití literálu	257
	Využití konstruktoru list(seq=())	257
	Sčítání a násobení	258
17.5	Generátorová notace seznamů	259
17.6	Modifikace seznamů	261
	Metody append() a extend()	261

Rizika práce s odkazy na proměnné objekty	262
Postupné budování seznamu	263
Přičítání jiných zdrojů.....	264
Indexace prvků seznamu	265
Metody pracující s indexy	266
index(value, start=0, stop=9223372036854775807, /) -> int.....	266
insert(index, object, /)	267
pop(index=-1, /) -> ?.....	267
remove(self, value, /).....	267
Metody pracující s celým seznamem.....	267
reverse(self, /)	267
sort(*, key=None, reverse=False)	267
17.7 Vícerozměrné seznamy	268
17.8 Souhrnný příklad	269
17.9 Anotace odkazující na seznamy.....	270
18 N-tice	271
18.1 Základní informace o n-ticích	271
18.2 Vytváření n-tic	271
Vytváření n-tic pomocí literálů	272
Využití konstruktoru tuple(seq=())	273
Sčítání a násobení	274
Přičítání n-tic	275
Balení a rozbalování n-tic.....	276
Prohazování proměnných.....	276
Hvězdičkové pravidlo	277
18.3 Generátorová notace n-tic	277
18.4 Problematika neměnnosti n-tic	278
Hešovatelné objekty	279
18.5 Přístup k prvkům n-tic.....	279
18.6 Sčítání seznamů a n-tic	280
18.7 Proměnné a neměnné prvky n-tice	280
18.8 Pojmenované n-tice.....	281
18.9 Anotace odkazující na n-tice	283
19 Množiny.....	284
19.1 Základní informace o množinách	284
19.2 Vytváření množin	284
Vytváření množin pomocí literálů	285
Vytváření množin pomocí konstruktoru set()	285
Hešová tabulka	285
Použitelné a nepoužitelné zdroje.....	286
Vytváření množin prostřednictvím množinových operací	287
union(*zdroj) a b 	288
intersection(*zdroj) a & b &	288
difference(*zdroj) a - b -	288
symmetric_difference(zdroj) a ^ b.....	289
19.3 Generátorová notace množin	289
19.4 Zmrazené množiny	289
19.5 Modifikace množin.....	290
Modifikace pracující s jedním prvkem	290
add(element).....	291
discard(element)	291
remove(element)	291
pop().....	291
Množinové operátory a sdružené operace.....	292

update (*zdroj) a = b 	293
intersection_update(*zdroj) a &= b &	293
difference_update(*zdroj) a -= b 	293
symmetric_difference_update(zdroj) a ^= b	293
Porovnávání množin	293
isdisjoint(množina)	293
a < b	293
issubset(množina) a <= b	294
issuperset (množina) a >= b	294
a > b	294
19.6 Anotace odkazující na množiny	294
20 Slovníky	295
20.1 Mapovací objekty a slovníky	295
20.2 Vytváření slovníků	296
Vytváření slovníků pomocí literálů	296
Vytváření slovníků pomocí konstruktoru dict()	297
Ekvivalence slovníků	298
Vytváření slovníků pomocí metody fromkeys()	298
20.3 Generátorová notace slovníků	299
20.4 Operace se slovníkem	300
Práce s hodnotami pomocí „indexace“ klíčem	300
Další metody pro práci s jednotlivými položkami	302
get(key, default=None, \)	302
pop(key, default=#, \)	302
popitem()	303
setdefault(key, default=None, \)	303
Modifikace slovníku daty ze zadaného zdroje	303
update(zdroj)	303
Operátor 	303
Sdružené přiřazení =	303
Slovník jako generátor	304
20.5 Pohledy	304
items()	304
keys()	304
values()	304
Pohledy jako zdroje dat	305
Operace s pohledy	305
20.6 Anotace odkazující na slovníky	306
21 Rozšíření definic funkcí	307
21.1 Předávání argumentů odkazem a hodnotou	307
Předání argumentu odkazem	308
21.2 Pomocná funkce gr()	309
21.3 Proměnný počet pozičních argumentů	310
Hvězdičkový parametr	310
Hvězdičkový argument	312
21.4 Proměnný počet pojmenovaných argumentů	313
Dvuhvězdičkový parametr	313
Dvuhvězdičkový argument	313
21.5 Stručný souhrn	314
Podivné chování	315
Použití v definicích literálů	316
21.6 Vestavěné funkce pracující s kontejnery	316
all(iterable)	316
any(iterable)	316

dir(object)	317
enumerate(iterable, start=0)	317
eval(expression, globals=None, locals=None, /)	317
exec(source, globals=None, locals=None, /)	318
frozenset(/iterable/)	318
filter(function, iterable)	318
globals()	318
locals()	318
len(c)	319
max(iterable, */, key, default/)	319
max(arg1, arg2, *args[, key])	319
min(iterable, */, key, default/)	319
min(arg1, arg2, *args[, key])	319
reversed(seq)	319
slice(stop) slice(start, stop[, step])	319
sorted(iterable, *, key=None, reverse=False)	320
sum(iterable, /, start=0)	320
vars(object)	320
zip(*iterables)	320
22 Formátování stringů.....	321
22.1 Formátovací operátor %.....	321
22.2 Pokročilejší metody formátování	323
Metoda format() versus f-stringy	323
Formátovací string.....	323
Formát nahrazovacího pole	324
Formát nahrazovaného textu.....	325
Konverze	326
22.3 Specifikace formátu.....	327
Počet zabraných pozic	327
Přesnost.....	328
Typ hodnoty	329
Stringy	329
Celočíselné hodnoty	329
Numerické hodnoty	330
Skupiny číslic	331
Alternativní formát a vedoucí nuly	333
Znaménko	334
Zarovnání a plnění	334
Vnořená nahrazovací pole	336
Formátování samodokumentujících se nahrazovacích polí.....	336
22.4 Příklad: Pascalův trojúhelník.....	337
22.5 Příklad: Trasovací funkce prSE() a prsIN()	338
Funkce prSE()	338
Funkce prIN()	340
Použití	341
23 Operace s kontejnery.....	343
23.1 Proměnné objekty jako implicitní hodnoty parametrů	343
23.2 Kopírování	343
Mělké a hluboké kopie objektů	344
Zdánlivé kopie neměnných kontejnerů	345
Alternativní způsob tvorby mělkých kopií.....	345
Nebezpečí hlubokých kopií.....	345
23.3 Rozdělení doposud probraných kontejnerů	346
23.4 Přítomnost prvku v kontejneru.....	347

23.5 Řazení prvků posloupnosti.....	347
reversed(seq)	347
sorted(iterable, *, key=None, reverse=False).....	348
23.6 Vykrajování (slicing).....	349
23.7 Indexování a vykrajování u rozsahů	350
23.8 Nahrazování hodnot	351
23.9 „Úprava“ neměnných objektů	352
24 Práce se soubory	353
24.1 Soubory: bleskové opakování.....	353
Soubor, souborový systém, cesta	354
Absolutní a relativní cesta	354
Substituované disky ve Windows.....	355
24.2 Práce se soubory v <i>Pythonu</i>	355
Starší koncepce souborů v jazycích C nebo Pascal.....	355
Novější koncepce datových proudů	356
Koncepce <i>Pythonu</i>	356
Cesty – soubory – proudy.....	356
Shrnutí používané terminologie.....	357
Soubor (anglicky file)	357
Složka	357
Cesta (anglicky path) – PLO.....	357
FLO (file like object), datový proud.....	357
24.3 Dva způsoby práce se souborovým systémem.....	358
24.4 Moduly <i>os</i> a <i>os.path</i>	359
24.5 Pracovní složka.....	360
24.6 Skládání a rozkládání cest	361
24.7 Vytváření a mazání složek	363
Mazání.....	363
24.8 Získání informací o souborech.....	364
24.9 Zápis a čtení dat.....	365
Problematika kódování ve Windows	365
Otevírání souborů a datových proudů	365
Zápis dat, splachování a zavírání proudů a přidružených souborů.....	368
24.10 Konstrukce <i>with</i> a správce kontextu	370
24.11 Čtení ze souborů.....	371

Část D Objektově orientované programování 373

25 Základy OOP	374
25.1 Předehra	374
Kdy se OOP začíná vyplácet	375
Různé pohledy na OOP	376
25.2 Základní princip OOP	376
25.3 Objekty a jejich atributy	377
Terminologická vsuvka.....	378
25.4 Třídy a jejich instance	379
Třída versus datový typ	379
Instance	380
25.5 Objekt třídy versus instance třídy	380
25.6 Atributy třídy versus atributy instancí.....	381
25.7 Zprávy	381
25.8 Metody	382
25.9 Dědění.....	382
Terminologie	382

LSP – substituční princip Liskové	383
Virtuální metody a jejich přebíjení	383
Zakrývání versus přebíjení.....	384
Polymorfismus	384
Rodičovský podobjekt.....	384
Násobné dědění a diamantový problém.....	385
Zobecňování.....	386
26 Třídy a jejich instance	387
26.1 Definice třídy a jejích atributů	387
Definice třídy je příkaz.....	389
26.2 Práce s atributy objektu	389
Získání a modifikace hodnoty atributu	389
Přidání a odebrání atributu	390
Nezveřejňované atributy	392
Třída jako parametr.....	392
26.3 Instance a práce s nimi	392
Textový podpis instance	393
Kvalifikace atributu třídy instancí	393
Tři druhy metod a použití příslušných dekorátorů.....	394
Metody instancí a parametr self	396
Převod metod na funkce a funkcí na metody	397
Vytváření instancí – konstruktor, alokátor, initor	397
Instanční metody třídy C2	398
Použití instančních a třídních atributů	400
Změny instančních a třídních datových atributů.....	401
Atribut třídy jako výchozí hodnota instančního atributu	403
Zavádění nových třídních funkčních atributů	404
Zavádění nových instančních funkčních atributů	405
26.4 Vestavěné třídy jsou nemodifikovatelné	406
26.5 Nutnost kvalifikace atributů.....	406
26.6 Některé speciální atributy – dunderý.....	407
__bases__	407
__class__	407
__dict__	407
__doc__	408
__mro__	408
__name__	408
__qualname__	408
__subclasses__ ()	409
Ukázky použití	409
26.7 Alternativní definice třídy.....	409
27 Jednoduché dědění.....	411
27.1 Vytváření potomka a jeho vlastnosti	411
27.2 Příklad.....	412
Třída LA	412
Třídy LB a LC.....	414
27.3 Jmenné prostory.....	414
Atributy a jmenové prostory tříd	414
Atributy a jmenové prostory instancí	415
27.4 Volání metod v hierarchii dědění	417
27.5 Initory se dědí	417
27.6 Hierarchie výjimek	418
27.7 Definice vlastní výjimky	419
27.8 Vestavěné funkce pracující s objekty	420
callable(object)	420

	classmethod(method, *args, **kwargs)	420
	delattr(object, name)	420
	getattr(object, name[, default])	420
	hasattr(object, name)	421
	isinstance(object, classinfo)	421
	issubclass(class, classinfo)	421
	setattr(object, name, value)	421
	staticmethod(method, *args, **kwargs)	421
	super([type/, object-or-type//])	421
28	Násobné dědění.....	422
	28.1 Python a násobné dědění	422
	28.2 Příklad: jednoduchý diamant.....	423
	28.3 Analýza chování.....	424
	Přímé zadání volaného initoru	425
	Problémy s parametry a argumenty – hubnoucí parametr	427
	28.4 Virtuální metody.....	429
	28.5 Složitější hierarchie dědění a MRO	429
	Nerealizovatelná a následně opravená hierarchie	429
	Zásady specifikace MRO	430
	28.6 Komolení jmen a pseudosoukromé atributy	431
	28.7 Šablonová metoda.....	432
29	Vlastnosti, abstraktní třídy a kachní typování	435
	29.1 Atributy × vlastnosti	435
	29.2 Přímá definice vlastnosti	436
	Využití lambda-výrazů, atributy jen pro čtení	438
	Vztah vlastnosti k třídě a instancím	438
	29.3 Zadání vlastnosti pomocí dekorátoru.....	440
	Použití.....	441
	Shrnutí	443
	29.4 Abstraktní × konkrétní třídy a metody	443
	Princip abstraktních metod	443
	Abstraktní a konkrétní třídy v mainstreamových jazycích	444
	Formálně a skutečně abstraktní třídy v jazyce Python	444
	29.5 Definice skutečně abstraktní třídy	445
	Přebíjet musí i ti, kteří abstraktní metodu nepotřebují	446
	Definice potomka abstraktní třídy	446
	Proč definovat metodu použitou v šabloně.....	447
	Proč dekorovat metodu jako abstraktní	447
	Proč v hlavičce deklarovat předka ABC	447
	Shrnutí	447
	29.6 Abstraktní vlastnosti a statické a třídní metody	448
	29.7 Rozhraní × implementace; protokol.....	449
	Protokol	450
	29.8 Kachní typování	450
30	Další objektové konstrukce	451
	30.1 Výčtové typy – základy.....	451
	Trocha terminologie.....	451
	Výčtový typ definovaný voláním funkce	452
	Vlastnosti výčtového typu a jeho instancí.....	452
	Útroby výčtového typu.....	454
	Výčtový typ definovaný jako třída	455
	30.2 Další možnosti výčtových typů	456
	Přezdívký hodnot.....	456
	Automatické přiřazení obalovaných hodnot.....	458