

Obsah

1 Úvod	9
2 Programovací paradigmaty a programy	14
2.1 Programovací paradigmaty	14
2.1.1 Imperativní programování	14
2.1.2 Deklarativní programování	15
2.1.3 Implementace imperativního paradigmatu	15
2.2 Interaktivní a dávkové aplikace	16
2.2.1 Dávkový program	16
2.2.2 Interaktivní program	17
2.2.3 Tvorba dávkových a interaktivních programů	17
2.2.4 Interpretace a kompilace	18
2.2.5 Vlastnosti interpretovaného a kompilovaného jazyka	18
3 Elementární algoritmy	22
3.1 Základní prvky imperativního paradigmatu	22
3.2 Vyšší celky – elementární algoritmy	23
3.2.1 Přechzení celého vstupu	23
3.2.2 Přechzení vstupu s koncovou hodnotou	23
3.2.3 Přechzení známého počtu hodnot	24
3.2.4 Zpracování dvojic hodnot	24
3.2.5 Vyhledání extrému	25
3.2.6 Zjištění počtu významných hodnot	26
3.2.7 Zjištění agregované veličiny	26
3.2.8 Dvouprůchodové algoritmy	26
3.3 Aplikace elementárních algoritmů	27
3.4 Kultura zápisu zdrojového textu	29
3.5 Opakovací příklady	34
3.5.1 Příklady bez cyklu	35
3.5.2 Zpracování řady hodnot	35
3.5.3 Strukturované datové typy	38
4 Konkrétní datové typy	50
4.1 Jednoduché datové typy	50
4.1.1 Ordinální datové typy	51
4.1.2 Celé číslo	51
4.2 Bitové operace s celočíselnými hodnotami	52
4.2.1 Znak	53
4.2.2 Výčet	54
4.2.3 Interval	54
4.2.4 Reálné datové typy	55

4.3	Strukturované datové typy	55
4.3.1	Pole	55
4.3.2	Řetězec	56
4.3.3	Množina	57
4.3.4	Záznam	58
4.3.5	Variantní záznam	58
4.3.6	Soubor	59
4.3.7	Datový typ podprogram	61
4.4	Konstanty s udaným typem	63
4.5	Proměnné s počáteční hodnotou	64
4.6	Příklady na procvičení	66
5	O datových typech a jejich kontrole	70
5.1	Typová kontrola a přetypování	70
5.1.1	Typová kontrola	70
5.1.2	Přetypování	71
5.1.3	Typová konverze	73
5.2	Obecné datové typy	73
6	Vstupy a výstupy v textové podobě	78
6.1	Standardní procedury <code>read</code> a <code>write</code>	78
6.2	Koncept <code>get/put, buf</code>	79
6.3	Čtení dat různých typů	81
6.3.1	Datový typ <code>char</code>	81
6.3.2	Datový typ <code>string</code>	82
6.3.3	Celočíselný datový typ	82
6.3.4	Reálný datový typ	84
6.4	Funkce <code>SeekEof, SeekEoln</code>	85
6.5	Vstupy a výstupy řádků, výstupy s formátem	85
6.6	Příklady na procvičení	87
7	Dynamické datové struktury	89
7.1	Statické a dynamické datové struktury	90
7.1.1	Statické, nebo dynamické?	90
7.1.2	Výhody a nevýhody dynamických datových typů	90
7.2	Datový typ ukazatel	91
7.2.1	Ukazatel	91
7.2.2	Ukazatele v Pascalu	92
7.2.3	Definice datového typu ukazatel	92
7.2.4	Práce s ukazateli a dynamickými proměnnými	93
7.3	Řešený příklad	95
7.4	Dynamické struktury	97
7.5	Lineární seznam	98
7.6	Základní operace s lineárním seznamem	99

7.6.1	Inicializace	99
7.6.2	Zjištění prázdnoti	99
7.6.3	Vytvoření prvku	99
7.6.4	Vložení prvku do seznamu	100
7.6.5	Průchod seznamem	100
7.6.6	Vkládání prvků na různá místa	101
7.6.7	Hledání	101
7.6.8	Zjištění počtu prvků	101
7.6.9	Odstranění prvku	102
7.6.10	Další operace nad seznamy	102
7.7	Kruhový seznam	102
7.8	Obousměrný seznam	104
7.9	Nelineární dynamické struktury	106
7.10	Příklady na procvičení	111
8	Abstraktní datové typy	116
8.1	Pojem abstraktního datového typu	116
8.1.1	Popis množiny hodnot	117
8.1.2	Popis množiny operací	117
8.1.3	Diagram signatury	117
8.2	Sémantika operací	118
8.3	Implementace abstraktních datových typů	120
8.4	Návrh vlastního abstraktního datového typu	121
8.4.1	Implementace dynamickým lineárním seznamem	122
8.4.2	Implementace polem	126
8.4.3	Shrnutí implementace	129
8.5	Příklady na procvičení	130
9	Typické abstraktní typy dat	134
9.1	Přehled typických abstraktních datových typů	134
9.2	Lineární seznam a jeho varianty	135
9.2.1	Jednosměrný seznam	135
9.2.2	Obousměrný seznam	136
9.2.3	Kruhový seznam	136
9.3	Aktivní seznam	136
9.4	Zásobník	137
9.4.1	Vlastnosti zásobníku	137
9.4.2	Operace se zásobníkem	138
9.4.3	Implementace zásobníku	138
9.5	Fronta	141
9.5.1	Vlastnosti fronty	142
9.5.2	Operace s frontou	142
9.5.3	Implementace fronty dynamickou strukturou	142
9.5.4	Implementace fronty polem	144

9.5.5	Prioritní fronta	145
9.6	Graf	145
9.7	Strom	146
9.7.1	Vlastnosti stromu	147
9.7.2	Operace se stromem	147
9.7.3	Metody procházení stromu	147
9.7.4	Implementace stromu	148
9.8	Řídké pole	150
9.9	Příklady na procvičení	153
10	Rekurze	158
10.1	Princip rekurze	158
10.1.1	Co je rekurze	158
10.1.2	Typy rekurzí	160
10.1.3	Použití rekurze	161
10.2	Příklady rekurzivních algoritmů	161
10.2.1	Faktoriál	161
10.2.2	Výpočet x^n	162
10.2.3	Fibonacciho posloupnost	163
10.2.4	Stromové algoritmy	164
10.2.5	Quick sort	165
10.2.6	Hanojské věže	166
10.2.7	Efektivita rekurzivních algoritmů	167
10.3	Příklady na procvičení	169
11	Strukturované programování, moduly	172
11.1	Programové moduly	172
11.1.1	Použití modulu v programu	173
11.1.2	Struktura modulu	173
11.1.3	Příklad implementace abstraktního typu zásobník řetězců	174
11.1.4	Překlad modulů	176
11.1.5	Cirkulární reference modulů	176
11.1.6	Moduly a dokumentace	177
11.2	Příklady na procvičení	178
12	Objektové programování	183
12.1	Pojem objektu	183
12.1.1	Objektová implementace abstraktních datových typů	183
12.1.2	Pojem objektu	184
12.1.3	Definice objektu	184
12.2	Vlastnosti objektů	186
12.2.1	Zapouzdřenost	186
12.2.2	Dědičnost	187
12.2.3	Polymorfismus a kompatibilita objektů	188

12.3	Manipulace s objekty	189
12.3.1	Statické metody objektů	189
12.3.2	Virtuální metody objektů	189
12.3.3	Rozšíření procedur <code>new</code> a <code>dispose</code>	191
12.3.4	Viditelnost objektů	192
12.3.5	Objektové knihovny	193
12.4	Příklady na procvičení	194
13	Programy a operační systém	199
13.1	Způsoby získávání dat	199
13.1.1	Standardní vstup a výstup	199
13.1.2	Hodnoty získané z příkazového řádku	200
13.1.3	Hodnoty získané z proměnných prostředí	201
13.1.4	Okna a dialogové prvky	201
13.2	Práce se soubory na disku	201
13.3	Příklad vazby na operační systém	201
13.4	Příklady na procvičení	205
14	Algoritmy a jejich hodnocení	207
14.1	Časová a prostorová složitost algoritmů	207
14.1.1	Složitost algoritmů	207
14.1.2	Způsob zápisu	208
14.1.3	Varianty složitosti	208
14.1.4	Stanovení složitosti	209
14.1.5	Porovnání složitostí programů	209
14.2	Přehled typických složitostí	210
14.2.1	Třídy složitostí	210
14.2.2	Typické příklady časové složitosti algoritmů	211
14.2.3	Snižování složitosti a efektivita algoritmů	211
14.3	Příklady na procvičení	213
15	Vyhledávání	215
15.1	Vyhledávání v lineárních strukturách	215
15.1.1	Základní úloha	215
15.1.2	Sekvenční vyhledávání v lineární struktuře	215
15.1.3	Hledání v uspořádané lineární struktuře	217
15.2	Vyhledávání v ostatních strukturách	218
15.2.1	Binární vyhledávací strom	218
15.3	Tabulka s rozptýlenými hesly	219
15.4	Příklady na procvičení	224
16	Řazení	228
16.1	Vlastnosti řadicích metod	228
16.1.1	Klasifikace algoritmů	229

16.2	Výběrové metody	229
16.2.1	Přímý výběr	229
16.2.2	Bublinové řazení	230
16.2.3	Řazení hromadou	231
16.3	Vkládací metody	233
16.3.1	Přímé vkládání	233
16.3.2	Řazení binárním stromem	234
16.3.3	Řazení množinou	234
16.3.4	Řazení rozptylováním	234
16.4	Rozdělovací a slučovací metody	235
16.4.1	Metoda Quick sort	235
16.4.2	Princip slučování	236
16.4.3	Jiné principy	237
16.5	Porovnání časových složitostí řadicích metod	238
16.5.1	Experimentální zjištění	238
16.5.2	Charakteristiky řadicích metod	240
16.6	Příklady na procvičení	243
17	Závěr	246
17.1	Řešené souhrnné úlohy	246
18	Literatura	258