

Obsah

1. Základní pojmy technologie programování	8
1.1. Historická charakteristika	8
1.2. Charakteristika chybovosti programového vybavení	9
1.3. Klasifikace programového vybavení	9
1.4. Softwarová krize	10
1.5. Vznik technologie programování	10
1.6. Typické postupy tvorby programů	10
1.6.1. Klasický životní cyklus programu	10
1.6.2. Tvorba programů s použitím prototypu	11
1.6.3. Technologie čtvrté generace (Four Generation Techniques - 4GT)	13
1.7. Závěr	13
2. Plánování projektu	14
2.1. Výpočetní systém	14
2.2. Technologie programování	14
2.3. Základní pojmy plánování programového projektu	14
2.4. Metrologie programátorské produktivity a kvality	16
2.5. Odhady a prognózy	17
2.6. Lidský činitel	18
2.7. Koupit či vyrobit?	18
2.8. Plánování organizace pracovního týmu	19
3. Analýza zadání	20
3.1. Základy analýzy zadání	20
3.1.1. Principy analýzy	20
3.1.2. Objektově orientovaná analýza	21
3.1.3. Analýza prováděná na modelu - prototypu	21
3.1.4. Specifikace	22
3.1.5. Reprezentace zadání	22
3.1.6. Ukázka obsahu specifikace	22
3.1.7. Revize specifikací	23
3.2. Metody analýzy zadání	23
3.2.1. Metody analýzy tokem dat	23
3.2.2. Metody analýzy datovou strukturou	24
3.2.3. Automatizované metody analýzy zadání	28
4. Návrh programu	30
4.1. Základní pojmy návrhu	30
4.1.1. Zjemňování	30
4.1.2. Architektura programu	30
4.1.3. Struktura programu	30
4.1.4. Struktura údajů	30
4.1.5. Programové procedury	31
4.1.6. Modularita	32
4.1.7. Abstrakce	32
4.1.8. Ukrývání informace	32
4.2. Návrh na principu toku dat	32
4.2.1. Tok transformací (funkční dekompozice)	33
4.2.2. Tok transakcí	36
4.3. Návrh na principu struktury dat	38
4.3.1. Jacksonova metoda	39
4.4. Objektově orientovaný návrh	43
4.4.1. Principy objektově-orientovaného návrhu	43
4.4.2. Metody OOD	44
4.4.3. Příklad OOD	45
4.4.4. Shrnutí OOD	48

Technologie programování v jazyce MODULA-2

1. NOVÉ PRINCIPY MODULY - 2	.51
1.1. MODULY	.51
1.2. SAMOSTATNĚ KOMPILOVANÉ MODULY	.53
1.3. MODULY PROGRAMU A PODPROGRAMU	.56
1.4. POMOCNÉ MODULY	.57
1.5. KNIHOVNA MODULŮ	.58
1.6. PROGRAMOVÁNÍ NA ÚROVNI BLÍZKÉ STROJI	.59
1.7. KOPROGRAMY A PŘERUŠENÍ	.60
1.8. PROMĚNNÉ TYPY PROCEDURA	.64
2. ODCHYLKY MODULY-2 OD PASCALU	.66
2.1. SLOVNÍK	.66
2.1.1. IDENTIFIKÁTORY	.66
2.1.2. SYMBOLY A REZERVOVANÁ SLOVA	.66
2.1.3. KOMENTÁŘE	.66
2.2. KONSTANTY	.66
2.2.1. TYP INTEGER	.67
2.2.2. TYP REAL	.67
2.2.3. TYP ZNAK	.67
2.2.4. ŘETĚZCE	.67
2.2.5. MNOŽINY	.67
2.3. TYPY	.68
2.3.1. INTEGER	.68
2.3.2. CARDINAL	.68
2.3.3. REAL	.68
2.3.4. BOOLEAN	.69
2.3.5. CHAR	.69
2.3.6. TYP DEFINOVANÝ VÝČTEM	.69
2.3.7. INTERVAL	.69
2.3.8. POLE	.69
2.3.9. ZÁZNAMY	.70
2.3.10. MNOŽINY	.70
2.3.11. UKAZATEL	.71
2.4. VÝRAZY	.71
2.4.1. FUNKCE	.71
2.4.2. OPERÁTORY	.72
2.5. PŘÍKAZY	.72
2.5.1. PŘÍŘAZOVACÍ PŘÍKAZ	.72
2.5.2. PŘÍKAZ VOLÁNÍ PROCEDURY	.73
2.5.3. PŘÍKAZ WHILE	.73
2.5.4. PŘÍKAZ IF	.73
2.5.5. PŘÍKAZ REPEAT	.74
2.5.6. PŘÍKAZ FOR	.74
2.5.7. PŘÍKAZ WITH	.74
2.5.8. PŘÍKAZ CASE	.74
2.5.9. PŘÍKAZY LOOP A EXIT	.75
2.5.10. PŘÍKAZ RETURN	.75
2.6. PROCEDURY A FUNKCE	.75
2.6.1. FUNKČNÍ PROCEDURY	.75
2.6.2. PARAMETR TYPU "OPEN ARRAY"	.75
2.7. STANDARDNÍ PROCEDURY	.76
2.8. BLOKY	.77
3. POMOCNÉ MODULY	.78
3.1. STANDARDNÍ V/V MODUL InOut	.78
3.2. VSTUP/VÝSTUP TEXTU MODUL Texts	.80
3.3. PRÁCE SE SOUBORY MODUL Files	.82
3.4. MODUL Terminal	.87

3.5. MODUL PŘIDĚLOVÁNÍ PAMĚTI Storage	.88
3.6. MODUL VOLÁNÍ PODPROGRAMU Program	.88
3.7. ŘETĚŽE MODUL Strings	.92
3.8. KONVERZE FORMÁTU MODULY - Conversions a RealConversions	.93
3.9. MATEMATICKÉ FUNKCE MODUL MathLib0	.94
4.1. MODULY	.95
4.2. DATOVÉ ABSTRAKCE	.96
4.2.1. DATOVÉ TYPY A DATOVÉ ABSTRAKCE	.96
4.2.2. PŘÍKLAD DATOVÉ ABSTRAKCE V MODULE-2	.96
4.2.3. PŘÍKLAD PRŮHLEDNÉHO A SKRYTÉHO ATD	.97
4.2.4. FUNKČNÍ ABSTRAKCE	.99
4.3. ABSTRAKCE BLÍZKÉ STROJOVÉ ÚROVNI	.99
4.4. STATICKÁ IMPLEMENTACE TYPICKÝCH ATD	104
4.4.1. ZÁSOBNÍK	104
4.4.2. FRONTA	105
4.4.3. MNOŽINA ZNAKŮ	107
4.5. DYNAMICKÉ DATOVÉ STRUKTURY	109
4.5.1. PŘÍKLAD POUŽITÍ ALLOCATE A DEALLOCATE	110
4.5.2. PŘÍKLAD DYNAMICKÉ PRŮHLEDNÉ IMPLEMENTACE ZÁSOBNÍKU	110
4.5.3. IMPLEMENTACE DYNAMICKÉHO GENERICKÉHO ATD ZÁSOBNÍK	111
4.5.4. NEPRŮHLEDNÁ IMPLEMENTACE DYNAMICKÉHO ATD FRONTA	112
4.5.5. IMPLEMENTACE ATD DYNAMICKÁ MATICE	114
4.6. PROCES JAKO ABSTRAKCE	117
4.6.1. PROCESY V MODULE-2	118
4.6.2. SYNCHRONIZACE PROCESŮ	119
4.6.3. KOMUNIKACE MEZI PROCESY	125
4.6.4. PŘÍKLADY NA APLIKACI PROCESŮ	129
4.7. OBJEKTOVĚ ORIENTOVANÝ NÁVRH PROGRAMOVÉHO SYSTÉMU	132
4.8. PŘÍKLAD PŘEDPROCESORU PRO IDENTIFIKÁTORY PROGRAMU ZAPSANÉHO V MODULE-2	133
Literatura	141

Technologie programování v jazyce Turbo Pascal 5.5

1. Principy objektivě orientovaného programování	142
1.1. Úvod	142
1.2. Kvalita programu a její složky	142
1.2.1. Šprávnost programu	143
1.2.2. Robustnost programu	143
1.2.3. Modifikovatelnost programu	143
1.2.4. Znovupoužití	143
1.2.5. Kompatibilita	143
1.2.6. Další složky kvality programu	143
1.2.7. Údržba programů	144
1.3. Modularita	144
1.3.1. Pět kritérií hodnocení návrhu modulů	144
1.3.2. Pět principů modularity	145
1.3.3. Otevřený a uzavřený modul	145
1.4. Různé pohledy na znovupoužití	146
1.4.1. Struktura modulu s ohledem na znovupoužití	146
1.4.2. Podprogramy	147
1.4.3. Balíky (packages) a moduly	147
1.4.4. Polymorfismus a generičnost	147
1.5. Cesta k objektivě orientované technologii programování	148
1.5.1. Funkce versus data	148
1.5.2. Objektivě orientovaný návrh	149
1.5.3. Nalezení objektů	149
1.5.4. Abstraktní typ dat jako popis třídy objektů	149

1.5.5.	Upřesnění definice objektivě orientovaného návrhu	150
1.5.6.	Sedm kroků na cestě k objektivě orientovanému návrhu	151
1.6.	Vlastnosti objektivě orientovaného návrhu	151
1.6.1.	Filosofie návrhu	151
1.6.2.	Hledání tříd	152
1.6.3.	Rozhraní a jeho techniky	152
1.6.4.	Dědičnost a její techniky	153
1.6.5.	Je lepší "zdědit" nebo "koupit" ?	153
1.7.	Závěr	153
2.	Objektivě orientované programování v Turbo Pascalu 5.5	154
2.1.	Objekty a dědění jejich vlastností	155
2.2.	Objekt a jeho použití	155
2.3.	Metody	156
2.3.1.	Obalení dat a metod v objektu	156
2.3.2.	Definice metod	156
2.3.3.	Rozsah platnosti metod a implicitní parametr Self.	157
2.3.4.	Objekty v modulových jednotkách Unit	158
2.3.5.	Změna a rozšiřování objektu	159
2.3.6.	Polymorfismus	161
2.3.7.	Kompatibilita objektů	162
2.3.8.	Úvaha o polymorfních objektech.	163
2.3.9.	Virtuální metody	163
2.3.10.	Rozdíl mezi procedurou a metodou	164
2.3.11.	Rozšiřitelnost objektů	168
2.3.12.	Statické versus virtuální	169
2.3.13.	Dynamické objekty	169
2.3.14.	Destruktory	170
2.3.15.	Příklad dynamicky vytvořených objektů.	171
3.	Definice jazyka Turbo Pascal 5.5	174
3.1.	Nové rezervované Identifikátory	174
3.2.	Typ objektu	174
3.3.	Kompatibilita vzhledem k přiřazení	176
3.4.	Zápis komponenty objektu	176
3.5.	Inicializace instance	176
3.6.	Konstanty typu object	177
3.7.	Vstupní bod metody	177
3.8.	Volání funkce	177
3.9.	Přiřazovací příkaz	177
3.10.	Příkazy procedury	177
3.11.	Příkaz case	178
3.12.	Příkaz with	178
3.13.	Deklarace metody	178
3.14.	Konstruktory a destruktory	178
3.15.	Parametr předávaný odkazem	179
3.16.	Rozšíření standardních procedur New a Dispose	179
3.17.	Symbody a direktivy podmíněného překladu	180
4.	Ladění objektivě orientovaných programů v TP5.5	181
4.1.	Objektivě orientované ladění v IVP	181
4.1.1.	Krokování a sledování metod	181
4.1.2.	Objekty v okně vyhodnocování	181
4.1.3.	Objekty uvnitř okna Watch	181
4.1.4.	Výrazy v příkazu Find Procedure	181
4.2.	Turbo Debugger 1.5	181
4.2.1.	Krokování a sledování metod	181
4.2.2.	Rozsah platnosti	182
4.2.3.	Vyhodnocovací okno Evaluate	182
4.2.4.	Sledovací okno Watch	183

4.2.5.	Okno objektové hierarchie	183
4.2.6.	Inspekční okno typu objektu T.D.	184
4.2.7.	Inspekční okno instance objektu	184
4.3.	Nová chybová hlášení	185
5.	Segmentace (Overlays)	186
5.1.	Ovládání vyrovnávací paměti segmentace	186
5.2.	Proměnné	187
5.2.1.	OvrTrapCount	187
5.2.2.	OvrLoadCount	187
5.2.3.	OvrFileMode	187
5.2.4.	OvrReadBuf	187
5.3.	Procedury a funkce	188
5.3.1.	OvrSetRetry	188
5.3.2.	OvrGetRetry	189
5.4.	Segmenty v souborech EXE	189
6.	Uvnitř Turbo Pascalu 5.5	190
6.1.	Vnitřní datová reprezentace objektů	190
6.1.1.	Tabulky virtuálních metod - TVM	190
6.1.2.	Standardní funkce TypeOf	191
6.1.3.	Volání virtuálních metod	192
6.2.	Konvence pro volání metod	192
6.2.1.	Konstruktory a destruktory	192
6.3.	Metody v jazyce symbolických instrukcí	193
6.4.	Zotavení se z chyby konstruktoru	195