

Stručný obsah

Předmluva	17
Úvod	19
Základy	35
Datum a čas	77
Aplikace	107
Kreslení v GDI+	123
Zpracování obrázků v GDI+	153
Windows Forms	225
Ovládací prvky Windows (Windows Controls)	295
Okno vlastností (PropertyGrid)	383
Souborový systém	437
Síť	495
Systém/WMI	513
Databáze	585
XML	637

Na přiloženém CD naleznete

Vědeckotechnické výpočty	723
Práce s geografickými daty	879
DirectX	955
Různé	981
Visual Basic 2008	1015
Visual Studio	1047
Zdroje na Internetu	1055
Základy dalších technologií	1061
Funkce API	1067

Obsah

Předmluva	17
Úvod	19
Základy	35
1 Číselné, znakové a řetězcové literály	35
2 Dvojkové, osmičkové nebo šestnáctkové vyjádření celých čísel	37
3 Převod řetězce ve dvojkové, osmičkové nebo šestnáctkové soustavě na číselnou hodnotu	37
4 Formátování číselných hodnot	37
5 Rozdílné formátování kladných a záporných čísel	38
6 Složená formátování	38
7 Definování poskytovatele formátu pro vlastní třídy	39
8 Výstupy ve formátech podle lokálních zvyklostí	40
9 Použití informací o lokálních nastaveních	42
10 Převod znakových řetězců na číselné typy	44
11 Kontrola, zda znakový řetězec obsahuje číselnou hodnotu	46
12 Největší a nejmenší hodnota číselného datového typu	46
13 Výpočet platných míst před desetinnou čárkou	48
14 Zalomení dlouhých jednořádkových textů	49
15 Obchodnické zaokrouhlování číselných hodnot	51
16 Kontrola, zda je v hodnotě typu Integer nastaven nějaký bit	52
17 Nastavení bitu v hodnotě typu Integer	53
18 Výmaz bitu v hodnotě typu Integer	54
19 Převod bitu v hodnotě typu Integer do určitého stavu	54
20 Změna bitu v hodnotě typu Integer	55
21 Zjištění nastavených bitů v hodnotě typu Integer	55
22 Dotaz na nenastavené bity v hodnotě typu Integer	56
23 Vytvoření booleovského pole z bitových informací hodnoty typu Integer	57
24 Sestavení hodnoty typu Integer z bitových informací booleovského pole	58
25 Konverze mezi 8bitovými, 16bitovými, 32bitovými a 64bitovými datovými typy	59
26 Stanovení bazového typu pro výčet	63
27 Vstup a výstup výčtových hodnot	64
28 Zjištění identifikátorů a hodnot výčtového typu	65
29 Kontrola, zda je číselná hodnota ve výčtu definována jako konstanta	66
30 Kontrola, zda se určitá hodnota výčtu vyskytuje v kombinaci hodnot	66
31 Vytvoření nabídek s hodnotami výčtu	67

32	Porovnávání a třídění objektů vlastních tříd	68
33	Binární vyhledávání v polích a seznamech	71
34	Převod řetězců na pole bajtů a naopak	73
35	Náhrada za konstantní řetězcová pole	75

Datum a čas 77

36	Práce s datem a časem	77
37	Přestupné roky	82
38	Výpočet dne v týdnu	82
39	Výpočet začátku kalendářního týdne	83
40	Určení počtu kalendářních týdnů v roce	84
41	Výpočet kalendářního týdne pro zadané datum	85
42	Výpočet data Velikonoc	87
43	Výpočet svátků	89
44	Znázornění svátků v ovládacím prvku kalendáře	99
45	Převod gregoriánského data na juliánské počítání dnů	101
46	Převod juliánského čísla dne na gregoriánské datum	103
47	Vstup a výstup data a času ve formátu ISO 8601	105

Aplikace 107

48	Vytvoření konfigurace aplikace pomocí Visual Studio	107
49	Uložení konfigurace pro připojení k databázi (se šifrováním a bez šifrování)	114
50	Dodatečné sekce v konfiguračním souboru	116
51	Čtení konfiguračního souboru machine.config	119
52	Nová nastavení aplikace	120
53	Centrální ošetření výjimek	121

Kreslení v GDI+ 123

54	Vytvoření obrysového písma	123
55	Výpis textu v kruhu a jeho otáčení	124
56	Vyplnění nápisu obrázkem v pozadí	129
57	Kreslení transparentního nápisu přes obraz	131
58	Blokové písmo pro zdůraznění nápisu	134
59	Kreslení textu s odsazeným stínem	135
60	Perspektivní zkreslení nápisu	137
61	Zjištění velikostí fontů pro přesné umístění nápisu	139
62	Vytvoření stínu pomocí maticových operací	142
63	Kreslení obdélníků se zaoblenými rohy	144
64	Vytvoření trojrozměrného písma	147
65	Trojrozměrné efekty a efekty osvětlení na strukturovaných obrazech pozadí	148

Zpracování obrázků v GDI+	153
66 Nakreslení obrázku	153
67 Zvětšení výřezu obrázku	156
68 Základní třída pro prezentaci obrázků	159
69 Horizontální a vertikální překrývání	164
70 Diagonální překrývání	169
71 Eliptické překrývání	172
72 Překrývání náhodným mozaikovým vzorem	174
73 Překrývání prolínáním	176
74 Maximalizace obrázků bez zkraslení	178
75 Zamezení blokování obrázkového souboru	182
76 Výběr složek s náhledy uložených obrázků	185
77 Otáčení a zrcadlení obrázků	198
78 Zjištění kodéru pro různé formáty souborů k ukládání obrázků	199
79 Uložení obrázků ve formátu JPEG	201
80 Uložení obrázků ve formátu GIF	202
81 Vytvoření miniatur pro webové stránky	203
82 Invertování obrázku	205
83 Převedení barevného obrázku na obrázek v odstínech šedé barvy	206
84 Další manipulace s obrázkem pomocí matice barev	207
85 Kopírování dat bitové mapy do pole	208
86 Kopírování pole do bitové mapy	210
87 Obecný filtr ostrosti	212
88 Ostrost podle Gausse	215
89 Ostrost pomocí Sobelova filtru	217
90 Ostrost pomocí Laplaceova filtru	218
91 Filtr Kirschův a filtr Prewittův	219
92 Filtr neostrosti Boxcar	219
93 Adaptivní ostrost	220
Windows Forms	225
94 Zobrazení okna bez titulkové lišty	225
95 Pohyb okna bez titulkové lišty	225
96 Poloprůhledná okna	226
97 Nepravidelná okna a jiné efekty s průhledností	227
98 Úvodní obrazovka	230
99 Zapouzdření dialogů	234
100 Zapouzdřený dialog s tlačítkem Použít	238
101 Základní třída Dialog	241
102 Validace zadaných dat	244

103	Vytváření snímků obrazovky	249
104	Třída TextViewer	254
105	Třída RTFTextViewer	257
106	Třída PictureBox	258
107	Prohlížeč HTML	261
108	Přetažení z jiných aplikací pomocí myši	264
109	Program pro analýzu přetažení z jiných aplikací	266
110	Zobrazení dat ze schránky	272
111	Export dat pomocí schránky	273
112	Export dat přetažením myší	275
113	Zachycení klávesy Enter v komponentě TextBox	277
114	Zkrácení cesty, aby se vešla do použitelné oblasti	279
115	Použití ovládacích prvků WPF ve Windows Forms	283
116	Přehrávání multimediálních souborů prostřednictvím WPF ve Windows Forms	289

Ovládací prvky Windows (Windows Controls)

295

117	Náhrada za pole ovládacích prvků z VB6	295
118	Seznam ovládacích prvků okna nebo procházení ovládacích prvků kontejneru	298
119	Určení událostí pro skupiny ovládacích prvků v návrhář	299
120	Vyhledání ovládacího prvku pomocí jeho jména	300
121	Nakreslení položek komponenty ListBox	300
122	Vícesloupcové pole se seznamem (ComboBox)	303
123	Základní třídy pro vámi definované ovládací prvky	306
124	Komponenta Label jako příklad možnosti rozšíření stávajícího ovládacího prvku	307
125	Uživatelský ovládací prvek jako kontejner pro jiné ovládací prvky	309
126	Zpřístupnění posuvníku uživatelského ovládacího prvku v návrhovém režimu	309
127	Uživatelské ovládací prvky a vlastnost Text	310
128	PanelGroupPictureBox – ovládací prvek pro všechny případy	311
129	Přiřazení ikony uživatelskému ovládacímu prvku	317
130	Nastavení průhlednosti ovládacího prvku v návrhář	318
131	Zachycení zpráv systému Windows	318
132	Ovládací prvek pro výběr adresáře	323
133	Ovládací prvek Průzkumník (Windows Explorer) vlastní konstrukce	334
134	Seřazení obsahu komponenty ListView ovládacího prvku Průzkumník	338
135	Ovládací prvek FolderBrowser se zaškrtačovacími políčky	345
136	Uživatelské ovládací prvky s neobdélníkovým obrysem	354
137	Přepočet pozice myši v kruhových ovládacích prvcích na úhel	356
138	Zachycení událostí myši v době návrhu	359
139	Ovládací prvek pro grafické znázornění časových úseků	360
140	Přidání nových časových intervalů za běhu programu pomocí myši	370

141	Posílání zpráv pomocí SendMessage	373
142	Určení zobrazení řádků ve víceřádkovém textovém poli	374
143	Zjištění počtu řádků víceřádkového textového pole	376
144	Zjištění indexu řádku z indexu znaku (víceřádkové textové pole)	377
145	Zjištění indexu prvního znaku řádku (víceřádkové textové pole)	378
146	Určení indexu prvního viditelného řádku víceřádkového textového pole	378
147	Výpočet indexu znaku z grafických souřadnic (víceřádkové textové pole)	378
148	Zjištění souřadnic znaku (víceřádkové textové pole)	379
149	Posouvání textu ve víceřádkovém textovém poli nahoru a dolů prostřednictvím kódu	380
150	Nastavení zářezek tabulátoru ve víceřádkové komponentě TextBox	381

Okno vlastností (PropertyGrid)

383

151	Základní atributy	383
152	Současné zobrazení vlastností více objektů	387
153	Zpracování neplatných hodnot	389
154	Výchozí hodnoty vlastností	390
155	Určení výchozí vlastnosti	391
156	Ochrana vlastností proti přepisu v ovládacím prvku PropertyGrid	391
157	Kombinování hodnot výčtů	392
158	Vnořené rozbalitelné vlastnosti	393
159	Rozbalovací seznamy s výchozími hodnotami textových vlastností	397
160	Vizualizace hodnot vlastností pomocí obrázků	399
161	Zobrazení vlastního rozbalovacího editoru	401
162	Zpracování vlastností pomocí modálního dialogu	404
163	Využití dialogu pro otevření souboru při zadávání vlastností	409
164	Zobrazení a zpracování seznamů	411
165	Nabízení akcí prostřednictvím inteligentních značek	417
166	Dynamické vytváření a vkládání vlastností	421
167	Zobrazování vlastností v různých jazycích (lokalizace)	429
168	Vložení nových karet	432

Souborový systém

437

169	Knihovna	437
170	Práce se systémovými složkami v prostředí .NET	440
171	Název knihovny či aplikace běžícího procesu	443
172	Kontrola existence složky	445
173	Vytvoření složky	446
174	Výmaz složky	447
175	Přejmenování a přesun složky	447
176	Kopírování složky	449

177	Velikost složky včetně podsložek	451
178	Kontrola existence souboru	454
179	Názvy souborů	455
180	Přejmenování a přesun souboru	456
181	Kopírování souboru	457
182	Určení verze souboru	459
183	Velikost souboru	461
184	Porovnání souborů	462
185	Jména dočasných souborů	466
186	Hledání souborů ve více složkách	468
187	Načítání informací o souborech pomocí File System Object	470
188	Načítání informací o discích pomocí FSO	471
189	Převod dat oddělených oddělovači do XML	473
190	Monitorování souborového systému	476
191	Atributy souboru	479
192	Určení částí cesty	483
193	Určení absolutní a zkrácené (kanonické) cesty	485
194	Zjištění relativní cesty	486
195	Určení typu souboru a související ikony	488
196	Zobrazování informací o souborech pomocí jazyka LINQ	491
197	Zobrazení počtu složek pomocí jazyka LINQ	493
Sít'		495
198	Převod adres ve formátu IPv4 do formátu IPv6	495
199	Převod adres ve formátu IPv6 do formátu IPv4	497
200	Určení IP adresy počítače	498
201	Vytvoření seznamu síťových adaptérů	500
202	Zobrazení seznamu sdílených složek	502
203	Spuštění aplikace Internet Explorer	504
204	Vytváření FTP připojení pomocí programu	505
System/WMI		513
205	Úvod	513
206	Jmenné prostory WMI	514
207	Třídy WMI	515
208	Kontrola instalace WMI	516
209	Informace z Biosu	517
210	Model počítače	521
211	Stav posledního spuštění	522
212	Letní/zimní čas	523

213	Doména, do níž je počítač zařazen	524
214	Role počítače v doméně	524
215	Jméno uživatele	525
216	Rozlišení monitoru	527
217	Typ monitoru	528
218	Rozlišení v palcích	529
219	Načtení informací o logických discích pomocí WMI	530
220	Načtení informací o fyzických discích	533
221	Přehled nainstalovaných programů	537
222	Přímé spuštění programu pomocí názvu	540
223	Nepřímé spuštění programu pomocí souboru	541
224	Načítání parametrů z příkazového řádku	542
225	Vytvoření seznamu běžících procesů pomocí WMI	544
226	Vytvoření seznamu běžících procesů pomocí System.Diagnostics	547
227	Seznam služeb	551
228	Vytvoření seznamu všech procesů pomocí LINQ	552
229	Spuštění služby	553
230	Pozastavení služby	556
231	Spuštění pozastavené služby	557
232	Zastavení služby	558
233	Přerušování běhu procesu	559
234	Monitorování výkonu/Čítače výkonu	562
235	Čtení informací z registru	565
236	Vytvoření klíče registru	567
237	Odstranění klíče z registru	568
238	Načítání informací o instalovaném operačním systému	569
239	Rychlost procesoru	574
240	Využití procesoru	575
241	Šířka sběrnice procesoru	576
242	Informace o procesoru	576
243	Informace o procesoru pomocí dotazu jazyka LINQ nad WMI	577
244	Odeslání elektronické pošty pomocí SMTP	581
245	Přihlašovací relace ve Windows	582

Databáze

585

246	Seznam dostupných SQL serverů	586
247	Výchozí přihlášení k SQL serveru	588
248	Přihlášení k SQL serveru s využitím uživatelského účtu Windows	589
249	Přihlášení k SQL serveru s využitím ověřování SQL serverem	590
250	Načtení seznamu databází serveru	591

251	Výběr databáze	591
252	Vytvoření seznamu tabulek databáze	592
253	Vytvoření seznamu polí tabulky	592
254	Jednoduché zálohování databáze	593
255	Jednoduchá obnova databáze	593
256	Vytvoření zálohovacího zařízení	594
257	Zálohování dat na zálohovací zařízení	596
258	Vytvoření seznamu zálohovacích zařízení	598
259	Obnova databáze ze zálohovacího zařízení	599
260	Vytvoření databáze	601
261	Vytvoření databázového skriptu v jazyce T-SQL	603
262	Vytvoření plánované úlohy	606
263	Vytvoření seznamu definovaných úloh	612
264	Indexy tabulek	614
265	Ukládání obrázků do tabulek	616
266	Načtení dat do komponenty DataGrid	619
267	Určení verze MDAC	620
268	Načítání dat z tabulek MS Excel jako z databáze	622
269	Připojení k databázi pomocí SMO	623
270	Kontrola stavu databázového serveru po síti	625
271	Načtení seznamu procesů a uživatelů databázového serveru pomocí SMO	627
272	Vytvoření nové tabulky pomocí SMO	630
273	Kontrola konzistence databázových tabulek	632
274	Protokolování aktivit SQL Serveru	633

XML

275	Vytváření XML souborů pomocí třídy XmlTextWriter	637
276	Načítání XML souborů pomocí třídy XmlTextReader	641
277	Ukládání obrázků a jiných binárních dat do XML souborů	645
278	Načítání obrázků a jiných binárních dat z XML souborů	647
279	Načítání XML souborů pomocí třídy XmlDocument	649
280	Vkládání, výmaz a změny uzlů pomocí třídy XmlDocument	651
281	Procházení XML dokumentů pomocí výrazů XPath	653
282	Dotazy XPath a jmenné prostory XML	658
283	Rychlejší vyhledávání pomocí tříd XPathDocument a XPathNavigator	662
284	Ovládací prvek XmlView, umožňující strukturované zobrazení XML souborů	665
285	Čtení zpráv z RSS kanálů	672
286	Nejdůležitější zprávy dne v uživatelském ovládacím prvku	678
287	Validace XML souborů	683
288	XSL transformace	689

637

289	XSL transformace s parametry	693
290	Příprava dodatečných funkcí pro XSL transformace	695
291	Paralelní používání XmlDocument a DataSet	697
292	Vytvoření hierarchie tříd z XML schématu	699
293	Serializace pomocí třídy XmlSerializer	701
294	Neznámý XML obsah při deserializaci pomocí třídy XmlSerializer	707
295	Serializace pomocí třídy SoapFormatter	708
296	Převod souboru ve formátu CSV do XML pomocí LINQ	711
297	Čtení XML souboru pomocí LINQ4XML	714
298	Změna XML elementů pomocí LINQ4XML	715
299	Přidávání a změna XML atributů pomocí LINQ4XML	716
300	Výmaz XML elementu a atributu pomocí LINQ4XML	717
301	Uložení XML dokumentu pomocí LINQ4XML	718
302	Načítání informací o XML dokumentu pomocí LINQ4XML	719

Na příloženém CD naleznete

Vědeckotechnické výpočty

723

303	Porovnávání desetinných čísel	723
304	Typově bezpečná práce s měrnými jednotkami	726
305	Definice délkových měř	742
306	Definice tříd pro vzdálenost a výšku	753
307	Definice plošných měř	756
308	Definice dutých měř	758
309	Definice času	760
310	Definice rychlosti	763
311	Definice teplot	765
312	Definice úhlových měř	766
313	Třída pro univerzální převody měrných jednotek	772
314	Soustavy lineárních rovnic o 2 neznámých	780
315	Soustavy lineárních rovnic o více neznámých	783
316	Vektorový počet ve 2D	789
317	Rozhraní pro zobrazování geometrických útvarů	797
318	Přímky	797
319	Výpočet průsečíku dvou přímek	803
320	Úsečky	807
321	Výpočet průsečíku přímky a úsečky	809
322	Výpočet průsečíku dvou úseček	812
323	Kružnice	814
324	Výpočet průsečíků dvou kružnic	816

325	Výpočet průsečíků přímky a kružnice	819
326	Výpočet průsečíků úsečky a kružnice	823
327	Uzavřené mnohoúhelníky	826
328	Náhrada kružnice mnohoúhelníkem	832
329	Výpočet průsečíků uzavřeného mnohoúhelníka a přímky	833
330	Výpočet průsečíků uzavřeného mnohoúhelníka a úsečky	835
331	Výpočet průsečíků uzavřeného mnohoúhelníka a kružnice	836
332	Výpočet průsečíků dvou uzavřených mnohoúhelníků	838
333	Generování různých tvarů pomocí vektorového počtu	839
334	Kreslení šipek s postupnou výplní zadní části	842
335	Úprava vzhledu stran mnohoúhelníka	843
336	Výpočet měřítka diagramu	846
337	Rozhraní pro datové zdroje s typově bezpečnými fyzikálními hodnotami	850
338	Jednoduchý T-Y diagram statických hodnot	853
339	Kontinuální T-Y diagram dynamických hodnot	866
340	Číslo π	875

Práce s geografickými daty

879

341	Zeměpisná délka a šířka	879
342	Výpočet vzdálenosti dvou bodů na zemském povrchu	886
343	Výpočet azimutu	888
344	Uživatelský ovládací prvek pro zadávání zeměpisných souřadnic	890
345	Rozšíření uživatelského ovládacího prvku PositionUC	897
346	Výšková data SRTM	899
347	Vyhodnocování dat z GPS přijímačů	904
348	Protokolování přijatých GPS dat	921
349	Zobrazení online mapy (Microsoft Virtual Earth)	925
350	Základní nastavení pro Yahoo! Maps	945
351	Základní nastavení pro Google Maps	949

DirectX

955

352	Vlastnosti grafického adaptéru	956
353	Kontrola podpory požadovaného režimu zobrazení	960
354	Matice DirectX	961
355	Jednoduchý torus	968
356	Komplexní grafické objekty	975

Různé

981

357	Přehrávání zvuku	981
358	Vytváření a přehrávání sinusových tónů	983
359	Přehrávání not	987

360	Přehrávání melodií	989
361	Řízení výstupů trasování a ladění pomocí konfiguračního souboru	995
362	Zobrazování ladicích a trasovacích informací v prvku typu TextBox	997
363	Ukládání ladicích a trasovacích informací do souboru	999
364	Zápis ladicích a trasovacích informací do protokolu událostí	1001
365	Vytvoření vlastního protokolu událostí	1002
366	Monitorování a čtení protokolu událostí	1005
367	Vytváření čítačů výkonu a načítání příslušných dat	1006
368	Přesné měření času	1010
369	Zpracování chybových hlášení API	1013

Visual Basic 2008 **1015**

Třídy – odkazy – objekty	1015
Struktury (hodnotové typy)	1017
Metody instancí	1017
Statické metody a statické proměnné	1019
Moduly	1019
Vlastnosti	1020
Dědičnost	1021
Generické datové typy	1025
Nullable (OfT)	1028
Anonymní typy	1029
Pole	1030
Seznamy (objektové)	1039
Generické seznamy	1043
Práce s podprocesy	1044
Kompatibilita s CLS	1045

Visual Studio **1047**

Odkládání textů do sady nástrojů	1047
Výchozí nastavení pro Option Strict	1047
Nastavení příkazů Imports pro celý projekt	1048
Ladění ovládacích prvků a tříd oken v režimu návrháře	1049
Přiřazení zástupce souboru k projektu	1051
Vytvoření a zpracování schématu XML pro daný XML soubor	1052
Navigace pomocí náhledu tříd	1052
Diagramy tříd	1053

Zdroje na Internetu **1055**

Weby o .NET	1055
Diskuzní skupiny	1057

Hledání pomocí Google	1058
Základy dalších technologií	1061
Stručný přehled WMI	1061
Základy XML DOM	1064
Funkce API	1067

Předmluva

„Dobrodružství pokračuje!“ V rukou právě držíte již třetí vydání knihy Visual Basic 2008 Codebook (pozn. českého vydavatele: tato informace se týká původního německého vydání). Celá platforma .NET se od verze 2.0 vyvíjela poněkud chaoticky, takže tato kniha se objevila poprvé až ve chvíli, kdy již byla dostupná verze .NET 3.5 SP1. A zatímco v případě verzí 1.0 a 2.0 se jednalo o skutečně nové verze, později se již něco podobného nedalo tak jednoznačně říci. Například verze 3.0 byla původně označována jako 2.1, podobně jako verze 1.1. Jenomže až teprve uvolnění verze 3.5 přineslo skutečně nové prvky. To znamená, že až tato verze by si vlastně zasloužila celé číslo verze. A .NET 3.5 SP1 opět přináší nové třídy a funkčnosti, a proto by měla nést označení .NET 3.6 (či spíše 3.1).

Zvyšování čísla verze je u platformy .NET provázáno také značným nárůstem jeho velikosti. Zatímco instalační balíček verze 1.0, uvolněný v lednu 2002, měl velikost okolo 20 MB, v případě verze 3.5 se již jedná o téměř 200 MB. Je tedy zřejmé, že firma Microsoft celou platformu značně rozšiřuje. A navíc k velikosti verze 3.5 musíte připočítat ještě velikost verze 2.0. Zatímco instalace verze 2.0 byla nezávislá na verzi 1.0, v případě verzí 3.x již tato nezávislost neplatí. Pokud bychom tedy chtěli formou zkratk popisat celou instalaci platformy .NET 3.5 SP1, pak by celá rovnice vypadala takto:

V3.5 SP1 (úplná) := V3.5 & V3.5 SP1 & V3.0 & V3.0 SP1 & V2.0 & V2.0 SP1

A to již byla oznámena další verze. Dne 29. 9. 2008 totiž firma Microsoft oznámila, že v roce 2010 uvolní .NET verze 4.0 a spolu s ním uvede na trh Visual Studio 2010 (neboli „Rosario“). Více informací najdete na adrese <http://www.microsoft.com/presspass/press/2008/sep08/09-29VS10PR.msp>. Přehled novinek, které v současnosti Microsoft pro tuto verzi plánuje, pak najdete na adrese <http://msdn.microsoft.com/en-us/vstudio/products/cc948977.aspx>. Jedno je tedy jisté: vývoj zůstává i nadále napínavý a stále se budete mít co učit.

Toto platí i pro aktuální verzi, již se zabývá tato kniha. Všechny recepty jsme prošli a přizpůsobili nové verzi, případně jsme do původní verze receptu přidali poznámky, týkající se nových možností. Takto jsme se rozhodli proto, abyste mohli i nadále pracovat či využívat starší verze. Někdy je použití starší verze .NET dokonce nutnost. A to nikoliv kvůli samotnému jazyku Visual Basic, ale především v důsledku požadavků či omezení, vznášených zákazníky či uživateli. Z vlastní zkušenosti jistě víte, že právě díky těmto požadavkům a omezením se každá nová verze prosazuje zpočátku velmi pomalu (pokud vůbec).

Byli jsme také dotazováni, jak na tom vlastně dnes jazyk Visual Basic je z hlediska využívání. Důvodem těchto dotazů byla zřejmě obava, že by se tazatel měl učit (a používat) nějaký programovací jazyk, který není nijak rozšířen. Mnohým se totiž jazyk C# jeví jako podstatně výkonnější a lepší. Jistě ale chápete, že odpovědět na takovou otázku je velmi těžké. A to především proto, že ani jeden z nás se neúčastnil jednání ve firmě Microsoft, během nichž se rozhodovalo o budoucnosti toho či onoho programovacího jazyka. Podíváte-li se ovšem na statistiky, běžně dostupné na Internetu, pak zjistíte, že pozice jazyka Visual Basic rozhodně není nijak špatná. I když i v tomto případě platí věta, že člověk by měl věřit pouze té statistice, jejíž výsledky sám zfalšoval (tento výrok je připisován Winstonu Churchillovi). Podíváte-li se například na aktuální statistiku TIOBE (<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>), pak zjistíte, že v prosinci 2008 se Visual Basic nacházel na pátém místě za jazyky Java, C, C++ a PHP. Jazyk C# je uveden na místě šestém. Pokud se týká trendů, pak ve srovnání s prosincem 2007 Visual Basic zaznamenal pokles, zatímco C# si „polepšil“. Navštívíte-li ovšem stránku *Programming Language Popularity* (<http://www.langpop.com>), pak v části *Normalized Comparison* uvidíte poněkud odlišné výsledky: Visual Basic se nachází na místě 11., a to za jazyky C, Java, C++, PHP, JavaScript, SQL,

C#, Python, Perl a Ruby. I přesto lze říci, že **Visual Basic zaujímá mezi ostatními programovacími jazyky významné místo.**

Je zřejmé, že s ohledem na rozmanitost jmenných prostorů a tříd jsme byli nuceni provést určitý výběr. Doufáme, že jsme výběrem témat vyhověli vašim požadavkům na takovouto knihu. Pokud budete v knize něco postrádat, objevíte chyby, nebo budete mít jiné podněty, napište nám. Zastihnete nás buď prostřednictvím vydavatelství, nebo přes níže uvedené internetové adresy. Nyní už jen zbývá, abychom vám popřáli mnoho zábavy při učení a mnoho zdaru při programování v jazyce Visual Basic 2008.

Autoři

Dr. Joachim Fuchs je samostatný vývojář softwaru, autor a docent, od roku 2001 je těžištěm jeho činnosti „Vývoj softwaru pomocí platformy .NET“. Své rozsáhlé vědomosti předává mimo jiné na seminářích, v diskusních skupinách Microsoftu a formou odborných článků pro různé časopisy. Dosažitelný je přes domovskou stránku <http://www.fuechse-online.de/beruflich/index.html>.

Andreas Barchfeld je vedoucí informačních technologií v jedné hamburské nemocnici. Má mnohaleté zkušenosti jako systémový a organizační programátor v oblasti operačních systémů Windows a Unix. Od vydání beta-verze 2001 platformy .NET se zabývá programováním na tomto poli. Těžiště jeho činností v této oblasti se nachází ve VB, C++ a v databázích. Dosažitelný je přes domovskou stránku <http://www.barchfeld-edv.com>.

Informace o knize a kontakt na autory

Prostřednictvím níže uvedených odkazů najdete aktuální informace o této knize a můžete s námi navázat kontakt.

<http://www.fuechse-online.de/vbcodebook/index.html>

<http://vbcodebook.barchfeld-edv.com>

Úvod

Od včerejška k dnešku

Toto malé ohlédnutí má za úkol objasnit historii vzniku a cíle jazyka Visual Basic 2008. Žádný strach, nepůjdeme zde do detailů, pouze načrtne okolnosti a záměry vývojářů jazyků Basic, Visual Basic a Visual Basic 2008.

10 LET A=4

Tak nebo podobně ještě mnozí znají první programy v jazyku Basic. První interprety Basicu existovaly na malých počítačích od firem Sinclair, Apple, Commodore atd. již koncem sedmdesátých let¹. Basic byl jednoduchý programovací jazyk, který mohl každý pochopit a jenž přinášel vše nezbytné pro běžnou potřebu. Zatímco jazyky Fortran, Cobol atd. bylo možno najít převážně na velkých počítačích, patřily interprety Basicu brzy ke standardnímu vybavení počítačů domácích.

S nezadržitelným rozvojem vyšších programovacích jazyků, jakými byly např. C, C++, Pascal a mnohé další, upadal Basic stále více do zapomnění. Novějšími jazyky bylo totiž možno vytvářet podstatně elegantnější a čistější programy než špagetové kódy Basicu.

On Error Goto

Když společnost Microsoft dodala prostřednictvím Windows do operačního systému grafické rozhraní, panovalo přesvědčení, že budoucnost programování je pevně spojena s jazykem C. Samotný operační systém Windows je z největší části programován ve standardním jazyku C. Rovněž první aplikační programy byly napsány v jazyku C.

Brzy se ukázalo, že pro programy systému Windows, které jsou řízeny především událostmi, byl jazyk C příliš rozvláčný. Tak nastal přechod k jazyku C++, což umožnilo dát k dispozici alespoň nejčastěji potřebné postupy v knihovnách tříd. Avšak také zacházení s knihovnamí tříd (jichž bylo zpočátku velmi mnoho) bylo vším jiným, než jednoduchou činností, a mnohé programátory to od dalšího vývoje programů pro Windows odrazovalo.

Vznikla potřeba jednoduchého jazyka s integrovaným vývojovým prostředím, které všechno přichystá, aby se „jen“ založilo okno s několika ovládacími prvky. Tak byl uveden v život jazyk Visual Basic. Základem byl Basic, jakožto jednoduchý programovací jazyk, spojený s (přinejmenším později) pohodlným vývojovým prostředím. Již verze 3 dokázala pokrýt mnohé případy použití, jež byly až do té doby doménou programování v C.

Co ve Visual Basicu chybělo, bylo možno doplnit přímým voláním funkcí z API. Visual Basic samotný byl dál trvale vyvíjen. Od verze 4 byly přidávány první objektově orientované nástavby, které však, bohužel, ještě ani ve verzi 6 nebyly úplné. První nástavba, které měla pomoci komponent v podobě souborů VBX učinit Visual Basic rozšiřitelným, byla opět brzo pozastavena a nahrazena komponentovým objektovým modelem (COM – později přejmenována na ActiveX).

¹ Jazyk Basic je ve skutečnosti ještě starší. Navrhli ho už v první polovině 60. let Kemeney a Kurz jako jazyk pro výuku programování a používal se na tehdejších sálových počítačích. (Pozn. odborného korektora.)

Prostřednictvím COM byl učiněn první krok ke stavebnicovému systému formou libovolně sestavitelných komponent. Komponenty, vytvořené pomocí Visual Basicu, lze také ještě dnes použít v jiných programech.

U produktů Office se stal jazyk Visual Basic for Applications (VBA) standardem pro automatizaci. Díky rozšíření VBA vzrostl ještě význam Visual Basicu.

Try Catch Finally

V podobě .NET nyní společnost Microsoft konečně vytvořila rozsáhlou platformu, která přesahuje programovací jazyk a je použitelná pro téměř všechny oblasti vývoje softwaru. Stavební kameny koncepce .NET vznikly na základě zkušeností, shromážděných do té doby při programování pomocí jazyků C++, VB a Java.

Všechny programovací jazyky pracují s tímtež systémem typů. Definice jako CLS (Common Language Specification), CLR (Common Language Runtime), CTS (Common Type System), a všem jazykům společná kompilace do bajtového kódu (MSIL – Microsoft Intermediate Language), přeloženého některým z kompilátorů JIT (Just In Time) na příslušném cílovém systému, zabezpečují flexibilní zaměnitelnost komponent. Třídy, které jsou uloženy v knihovně tříd jazyka Visual Basic 2008, je možno začlenit do programu v C#, použít v něm, nebo dokonce děděním rozšířit – a to platí i obráceně. Takto vytvořenou knihovnu však umí využívat také jakýkoliv jiný programovací jazyk, respektuje-li zásady .NET (Eiffel.NET, Perl.NET atd.)

Jazyky samotné ztratily na významu. Programovací jazyk slouží pouze k převedení algoritmů do funkčních (běhu schopných) programů. Veškeré prostředí, tj. vytváření oken, přístupy na databáze, kreslicí operace a mnohé další záležitosti už nejsou součástí jazyka, ale poskytuje je samotná platforma .NET, ohromná knihovna tříd.

U starých verzí VB došlo – bohužel – také k pokusům o zabudování chybějící funkčnosti Basicu do jazyka Visual Basic. Takže i ve Visual Basicu 2008 se stále ještě nalézají staré přítěže v podobě funkcí, které v programovacím jazyku vlastně nemají co pohledávat, ale spadají do oblasti úloh operačního systému. Původní záměr, pomoci při přechodu z VB6 k Visual Basicu .NET tím, že se velká část starých funkcí Basicu dá k dispozici také ve Visual Basicu .NET, působí příliš často kontraproduktivně. Bez nápovědy lze sotva zjistit, zda a jak se dá nějaká metoda z programovacího jazyka modelovat v .NET.

Zatímco přechod od jazyka C k jazyku C++ spočíval hlavně v tom, že se na jazyk C „naroubovala“ objektově orientovaná nastavba, byl v podobě Visual Basicu .NET vyvinut jakoby nový jazyk, který má s VB6 a předchůdci (od tohoto okamžiku nazývanými VB Classic) společnou už jen malou část syntaxe. Předpokladem jak pro programování pod .NET, tak pro Visual Basic 2008, je pochopení objektově orientovaného programování (OOP). Dokonce i když můžete jednoduché úlohy s procedurálním postupem – který bohužel u programování pomocí klasického VB převládá – vyřídit podobným způsobem také pomocí Visual Basicu 2008, narazíte velmi rychle na omezení. Možnosti celé platformy využijete jen tehdy, budete-li zvládat nejdůležitější koncepce OOP (třídy – objekty – odkazy, dědění, rozhraní atd.). Třebaže jsou mnohé recepty realizovány ve formě funkcí, je nutno tyto funkce chápat jako součást některé nadřazené třídy.

Platforma .NET se důsledně rozšiřuje a trvale roste. Koncem roku 2005 vyšla nová verze platformy .NET 2.0 a spolu s ní mnohé nové třídy a funkce. Součástí .NET jsou nyní také generické datové typy. Novinky se nevyhnuly ani jazykům. Nové návrhové nástroje – designery – pro zdroje a konfigurační soubory automaticky generují kód, který mnohé postupy zjednodušuje. Speciálně pro Visual Basic

2008 existuje nový jmenný prostor My, který má za úkol pomoci zejména novým uživatelům rychle vyhledat potřebné metody.

I následující verze 3.0 a 3.5 jsou založeny na Common Language Runtime verze 2.0. Rozšíření jazyka, která jsou novinkou verze 3.5, jsou kompilátorem plně převáděna na verzi 2.0. K nejdůležitějším rozšířením verze 3.0 patří WPF (Windows Presentation Foundation) a WCF (Windows Communication Foundation), zatímco u verze 3.5 je to LINQ (Language Integrated Query).

Co lze pomocí jazyka Visual Basic 2008 realizovat a co nikoli

Protože platforma .NET vybavuje všechny jazyky stejnými možnostmi, obor nasazení jazyka Visual Basic 2008 se oproti jazykům VB Classic výrazně rozšířil. Kromě aplikací pro Windows a knihoven tříd nyní můžete pomocí Visual Basicu 2008 programovat i webové aplikace, webové služby a konzolové aplikace. Realizovatelné jsou dokonce i služby Windows.

Ve Visual Basicu 2008 lze rovněž vyvíjet software pro příruční osobní počítače (PDA apod.), které jsou vybaveny odpovídající verzí platformy .NET. Visual Studio pro tyto přístroje nabízí speciální testovací prostředí. Také .NET Compact Framework prošel dalším vývojem a v současné době je k dispozici také ve verzi 3.5.

Oblastí, kde vůbec nelze použít platformu .NET, jsou ovladače zařízení. Ty zůstávají, stejně jako předtím, doménou programování v C a pomocí jazyka Visual Basic 2008 je nelze realizovat.

Leccos bylo vykonáno i na přímo využitelných charakteristikách operačního systému. Například pomocí VB6 bylo možné psát programy, využívající multithreading, nebylo však možné tuto funkci ve vývojovém prostředí otestovat. To se příchodem Visual Basicu .NET změnilo.

V zásadě můžete pomocí Visual Basicu 2008 realizovat vše, co dokážete s pomocí jazyka C#. Na základě syntaktických rozdílů mezi jazyky se dají mnohé věci v tom či onom jazyku vyjádřit elegantněji. Nepře-konatelná omezení, jaká dříve byla pro VB6 ve vztahu k C++, již u jazyků platformy .NET neexistují.

Obsah knihy

Chcete-li napsat knihu o .NET, musíte si položit otázky, na jaké úrovni se má začít a jaký má sledovat cíl. Jak jsme se zmínili již v předmluvě, není tato kniha určena programátorům v klasických jazycích VB, kteří dosud nemají s platformou .NET žádné zkušenosti. Pro přestup od VB6 k Visual Basicu 2008 a vstup do objektově orientovaného programování už existuje rozsáhlá literatura. Nejdůležitější pojmy, spojené s objektově orientovaným programováním pomocí jazyka Visual Basic 2008, jsme objasnili v odkazové části knihy. Toto krátké shrnutí však nemůže osvětlit všechny detaily a nenahra-zuje literaturu pro studium OOP.

Cílem knihy je spíše na základě prakticky orientovaných receptů ukázat, jak efektivně se s třídami platformy .NET pracuje, jak se zachází s takovými technikami, jakými jsou GDI+, XML, ADO.NET, Windows Forms, Windows Controls, WMI atd., a – v neposlední řadě – jak se v případech nouze dostat k funkcím z API Windows, jestliže potřebná funkce (zatím) není v .NET k dispozici.

Tyto recepty, jež najdete hned za tímto úvodem, představují hlavní část knihy. Vyvinuli jsme cca 300 receptů z nejrůznějších oblastí, rozdělených do několika kategorií. Témata a formulace úloh jsme vybírali na základě vlastních praktických zkušeností, mnoha otázek, jež nám byly kladeny v seminářích, diskusích v diskusních skupinách a podnětů četných příspěvků na Internetu.

Každý recept nejprve vysvětluje formulaci úlohy a poté ukazuje řešení, jež se skládá z komentovaných výpisů programů, obrázků a – především – objasnění způsobu řešení. K většině receptů najdete na CD v knize projekty Visual Studia, pomocí nichž jsme kód testovali. Vytisknuté výpisy programů si můžete zkopírovat přes úložiště (Repository), nacházející se rovněž na CD. Nemusíte tedy nic přepisovat.

Co je nového na nové knize řešení v jazyce Visual Basic 2008?

Nuže, nejnapadnějším znakem nové série knih řešení je přirozeně jejich vzhled. Třetí vydání se svou úpravou opět přiblížilo „normálním“ knihám. Karton s pouzdem pro CD je pryč. Mnohé ovšem bylo vykonáno také na obsahu. Všechny recepty jsme přepracovali a vyhodnotili, zda jsou i nadále hodny doporučení, či zda by neměly být nahrazeny novými možnostmi platformy .NET. Mnohé recepty prvního a druhého vydání jsme přepracovali, takže nyní využívají nových charakteristik Visual Basicu a platformy .NET. Některá vylepšení receptů a opravy chyb jsme mohli provést díky diskusím se čtenáři.

Přirozeně jsme do knihy přidali i recepty nové. Lze říci, že téměř bez výjimky nové recepty najdete ve všech kapitolách. Ať už se jedná o kapitolu věnovanou XML, práci s databázemi, či o novou kapitolu „Zpracování geografických dat“, do níž jsme přesunuli několik receptů z druhého vydání a k nim přidali mnoho dalších.

Komu je tato kniha určena

Tato kniha se zaměřuje na vývojáře aplikací, kteří již mají zkušenosti s platformou .NET. Při vysvětlování receptů předpokládá dobré znalosti programování a znalosti objektově orientovaného programování. Všechny zdrojové texty jsou napsány v jazyce Visual Basic 2008, jinak by přece bylo celé téma pochybné. Ale i programátoři používající jiných jazyků, kteří do jisté míry ovládají syntaxi Visual Basicu 2008, mohou recepty použít a převést do svých jazyků.

Těžiště témat jednoznačně spočívá v programování pro Windows, neboť webové programování je již pokryto obdobnou knihou řešení pro ASP.NET. Ale mnoho užitečných receptů v této knize najdou i weboví programátoři. Dokonce i recepty pro GDI+ mohou být pro webové aplikace zajímavé, například když se na serveru musejí online vytvářet obrázky.

Předpoklady pro využití receptů

Všechny projekty, které naleznete na CD, byly vytvořeny pomocí Visual Studia .NET 2003 a Frameworku verze 1.1 pod operačním systémem Windows 2000. Vycházíme z toho, že vy sami s Visual Studiemi .NET pracujete, neboť to je standard pro programování ve Visual Basicu 2008.

Můžete ale použít také freeware programovacího prostředí SharpDevelop. Ovšem SharpDevelop nemůže ani zdaleka dosáhnout rozsahu Visual Studia.NET. Pro zvlášť náročné je pak ještě k dispozici možnost využití bezplatného editoru Notepad a následného spouštění příkazů kompilátoru z okna příkazového řádku.

Vývoj softwaru pomocí jazyka Visual Basic 2008 byste měli provádět pouze na operačních systémech Windows XP, Windows Vista, Windows 2003 Server a Windows 2008 Server, resp. na jejich následovnících. Visual Studio .NET 2008 ostatně pracuje jen s těmito operačními systémy.

Mnohé recepty, rozhodně ne však všechny, jsou funkční i v prostředí **Mono**. V době tisku této knihy byla aktuální verze 2.0, obsahující již mnoho vlastností platformy .NET 3.x (jako například LINQ). Jinak Mono 2.0 plně pokrývá veškeré vlastnosti a funkce .NET 2.0.

Časté otázky ke knize řešení v jazyce Visual Basicu 2008 a k programování pomocí jazyka Visual Basic 2008

Jak dostanu vytištěné zdrojové texty z receptů do své aplikace?

Na CD u knihy se nachází program úložiště (Repository), které je uspořádáno podle kategorií. Vyhledáte příslušný recept s požadovaným výpisem a zdrojový text pomocí schránky zkopírujete do své aplikace.

Mohu převzít recepty do své aplikace beze změny?

Pokud to bylo možné, sestavili jsme recepty tak, abyste je mohli převzít beze změny. V mnohých případech to ovšem možné nebylo, neboť u objektově orientovaného programování nelze například metody jednoduše vytrhnout z vazeb uvnitř tříd. U Windows Forms nebo Windows Controls nelze z důvodů nedostatku místa vytisknout kód pro generování ovládacích prvků atd. Také skutečnosti dané syntaxí, jako např. příkazy importu na začátku souboru, často brání tomu, aby bylo možno metody a třídy klepnutím myši kopírovat. Většinou se musejí na několika místech provést změny.

Celou řadu receptů jsme objasnili na základě příkladů. V těchto případech pochopitelně musíte kód příkladu pro své účely upravit.

Existují k receptům příkladové projekty?

Ano, pro téměř všechny recepty na CD najdete příkladové projekty, v nichž je popsán kód použit a názorně vyložen. Přitom některé recepty byly objasněny na souvisejícím příkladu a potom zahrnuty do projektu. Jen pro několik málo receptů, které si vystačí s kódem, jenž nestojí za zmínku (např. k diskusi o základních třídách pro ovládací prvky), žádné vlastní projekty neexistují.

Jak najdu k receptu příslušný projekt?

Recepty jsou na CD uloženy v adresářích, jejichž jména začínají příslušnými čísly projektů. Přiřazení tak nečiní žádné potíže.

Mohu příkladové projekty ihned použít?

Mělo by to jít. Mějte však na paměti, že projekt se musí nejdříve zkopírovat na pevný disk, protože překladač nemůže na CD založit žádné soubory ani adresáře.

Jak mohu kód z receptu začlenit do svých programů?

Kromě kopírování fragmentů kódu můžete v mnoha případech využít také hotové knihovny, které jsme pro většinu kategorií dali k dispozici. Musíte pouze na potřebnou knihovnu odkázat pomocí seznamu odkazů. Pro programátory jiných jazyků, např. C#, mají knihovny tu výhodu, že mohou kód použít, aniž by museli sáhnout zpět po zdrojovém kódu Visual Basicu 2008.

Programuji pomocí jazyka C#. Mohu tuto knihu rovněž využít?

Ano. Hlavním tématem knihy je programování v prostředí platformy .NET a nikoli ukázat, jak se ve Visual Basicu programuje nějaký cyklus. Ovládáte-li C#, a umíte-li se trochu vypořádat se syntaxí Visual Basicu 2008, budete moci receptů využít. Můžete si vybrat, buď si kód VB sami převeďte do C#, nebo prostě využijete projektů a jako knihovnu tříd jej začleníte do projektu v C#.

Programuji pomocí ASP.NET. Co mi tato kniha přinese?

Hlavním tématem knihy je programování aplikací pro Windows. Avšak i ty zahrnují mnohé oblasti, které nemají s rozhraním Windows co činit. Například přístupy k systému, výpočty dat, přístupy k souborům XML a na databáze, nebo operace se soubory jsou pro webové uživatele právě tak zajímavé jako pro programátory ve Windows. Zvláště recepty pro okno vlastností mohou být velmi užitečné, když budete navrhovat webové ovládací prvky (Web Controls), které byste chtěli předat dalším uživatelům.

Zabývá se kniha ASP.NET?

Nezabývá, tomuto jazyku se věnuje kniha řešení v ASP.NET, které ani nechceme dělat konkurenci. Webové aplikace a služby jsou pojednány jen do té míry, v jaké je lze využít v aplikacích pro Windows. Speciálními tématy ASP.NET se v této knize nezabýváme.

Potřebuji Visual Studio k programování pomocí VB?

Ano i ne. V podstatě potřebujete jen .NET Framework SDK a textový editor (a vlastně si můžete programy vyděrovat na děrné štítky). Ale pokud jste již pracovali s jazykem Visual Basic 5 nebo 6, pak byste určitě neradi postrádali komfortní vývojové prostředí. Navíc je nutné zdůraznit, že verze Express Edition vývojového prostředí je zdarma. Velmi dobře však můžete pracovat i ve výše zmíněném volně šiřitelném prostředí SharpDevelop v kombinaci s platformou .NET.

Potřebuji Visual Studio pro použití receptů?

Také zde platí ano i ne. Teoreticky je nepotřebujete, avšak prakticky ano. Všechny projekty byly ve Visual Studiu vytvořeny a pomocí Visual Studia je lze pochopitelně také otevírat a dále zpracovávat. Všechny potřebné funkce pro práci s projekty Visual Studio vám samozřejmě nabízí i SharpDevelop.

Chci vyvjet moduly pro MS Office. Mohu recepty z této knihy využít?

Opět platí ano i ne. Chcete-li pracovat v jazyku Visual Basic for Applications (VBA), pak recepty použít nemůžete. Rozhodnete-li se ovšem pro sadu nástrojů Visual Studio Tools for Office (VSTO), pak recepty využít můžete. Nicméně, v této knize nenajdete žádné podrobnější informace, týkající se programování aplikací integrovaných s MS Office. Více informací najdete na adrese <http://msdn.microsoft.com/en-us/office/aa905533.aspx>.

Mám jen VB6. Mohu také příklady upotřebit?

Ne. Programování ve Visual Basicu 2008 už nemá s programováním ve VB6 nic společného. Příklady jsou spustitelné pouze v prostředí Visual Basicu 2008.

Jsou použity moduly nebo jiné, z důvodů kompatibility převzaté staré techniky jazyka VB6?

Nejsou. Tyto techniky jsou pro přestupující programátory při učení se koncepci .NET spíše na překážku než ku prospěchu. Vše, co lze pomocí platformy .NET a technik objektově orientovaného programování realizovat, jsme také tak realizovali. Tam, kde to má smysl, používáme přirozeně pojmy z VB, jako např. Integer místo System.Int32. Také pro běžné přeměny typů používáme v mnoha případech metody jako CInt nebo CSng místo velmi rozvláčné syntaxe CType nebo DirectCast.

V podstatě jsme toho mínění, že bývalý programátor VB6 se musí také (nebo snad především) osvobodit od starých zátěží klasického VB a že by neměl pracovat proti duchu koncepcí .NET. Pro procedurální programování, v němž se často pracovalo s globálními proměnnými a metodami, zde není místo.

Jsou probrány všechny tematické oblasti .NET?

Jestli si při tomto dotazu potají myslíte, „že přece ano“, pak byste měli vědět, že takovou knihu byste neunesli. Ne, platforma .NET zahrnuje tolik oblastí, že je zcela nemožné vše v jedné knize – byť jen přibližně – popsat. Vyhledali jsme témata, která mají – podle našeho názoru – pro programátory ve Visual Basicu význam a/nebo se ukázala jako relevantní v praxi, diskusních skupinách atd.

Jsou v knize detailně vysvětleny techniky, které jsou platformě .NET cizí, jako např. XML?

Nejsou, také zde u knihy narážíme na omezení. Vysvětlujeme, jak s těmito technikami můžete v .NET zacházet. Avšak např. základy XML nemůžeme podrobně dokumentovat, k tomuto účelu existuje dostatek literatury, v knihách i online.

Používá se i podprocesů (Multithreading)?

Jak jsme již uvedli výše, použití použití podprocesů se ve Visual Basicu 2008 podstatně zjednodušilo. Výslovně ovšem varujeme před používáním podprocesů bez důkladných znalostí jeho základů. Právě v oblasti aplikací pro Windows je mnoho případů, v nichž mohou nezkušení programátoři v podprocesech tápat. Podprocesy jsou značně komplikovanější, než by se na první pohled zdálo.

Podprocesy jsme použili jen ve výjimečných případech, kdy k tomu byl opravdu důvod.

Na jakých platformách běží programy vyvinuté v .NET?

Teoreticky na všech platformách, pro které existuje .NET Framework. To znamená na platformách Windows 98SE, Windows ME, Windows NT 4.0, Windows 2000, Windows XP, Windows Vista, Windows Server 2003 a Windows Server 2008. Ovšem pro Windows 98/ME existuje velmi mnoho omezení a podobně to vypadá u Windows NT. Úplná implementace platformy .NET existuje teprve pro Windows 2000 a vyšší. K platformám, které byly vydány před Windows 2000, se společnost Microsoft chová macešsky, a je nutno počítat s velkými problémy.

Na jakých platformách fungují příklady z knihy?

Vyvíjeli a testovali jsme pod systémem Windows XP. Programy by se měly chovat stejně pod Windows 2000 a Windows Server 2003. Na starších platformách některé příklady (např. s transparentními efekty) s největší pravděpodobností fungovat nebudou. Operační systémy Windows 98 a ME jsou konstruovány jinak než aktuální operační systémy, pro něž se s .NET počítá. Mnohé třídy .NET předpokládají skutečnosti, které staré systémy nepodporují nebo nepodporují úplně.

Fungují recepty také v prostředí platformy .NET 1.0, resp. 1.1 a s Visual Studiem .NET 2002, 2003, resp. 2005?

Z největší části pravděpodobně ano, my jsme to ale nevyzkoušeli. Ve zdrojovém kódu jsou případně nutné malé změny u cyklů typu `For`, neboť jsme zpravidla využili možnosti definovat proměnné cyklu v příkazu `For`. Ve výjimečných případech se může stát, že se upotřebí rozšíření .NET, která nejsou ve verzi 1.0 obsažena. Příkladem je enumerace `Environment.SpecialFolder`, která je ve verzi 1.1 výrazně rozsáhlejší.

Recepty, které využívají nové vlastnosti .NET 3.5 nebo Visual Studia 2008, se pochopitelně nedají použít na starších platformách. Ve verzích předchůdců například neexistují generické seznamy nebo noví průvodci Visual Studia 2008.

Fungují recepty také na Compact Frameworku na přenosném počítači (PDA)?

Částečně. Problém je v tom, že Compact Framework i ve verzi 3.5 dává k dispozici jen malou část tříd, metod, vlastností atd. Velkého bratra. Bude tak mnoho receptů, které lze bez problémů využít na Compact Frameworku, budou však také jiné, jež se buď musejí upravit, nebo na PDA vůbec použitelné nebudou.

Jaké předpoklady musejí být splněny pro příklady databází?

Pro příklady databází potřebujete SQL Server 2000, SQL Server 2005 nebo SQL Server 2008, přičemž varianta databázového serveru nehráje žádnou roli. Postačí dokonce i MSDE (2000) nebo SQL Server Express Edition (2005 či 2008).

Jaké předpoklady musejí být splněny pro příklady DirectX?

Vyvinuli a testovali jsme příklady s DirectX, verze 9.0c z října 2005. Pro převzetí velké části výpočtů z hardwaru by měla grafická karta podporovat DirectX 9.0. V opačném případě provádí všechny výpočty procesor počítače, čímž se může rychlost provádění výrazně snížit.

Tuto kapitolu jsme v knize ponechali, i když dnes je již dostupné rozhraní DirectX 10. Nicméně, na základě svých zkušeností si troufáme říct, že většího rozšíření se tato verze DirectX dočká až po nějaké době.

Používá se jmenný prostor My, resp. co se z něj má zachovat?

Především: Ano, používáme jej tam, kde to má smysl.

Podle záměru Microsoftu má jmenný prostor My zjednodušit programování ve Visual Basicu 2008 pomocí Visual Studia 2008. My je ovšem předmětem velmi kontroverzní diskuse, neboť ne každý pokládá tuto novinku za přednost. Pomocí jmenného prostoru My získáte přístup k některým méně používaným metodám a vlastnostem vybraných tříd .NET. Také přístup k prostředkům a nastavením aplikací probíhá ve Visual Basicu 2008 často přes jmenný prostor My.

Je ovšem nasnadě, že pomocí jmenného prostoru My lze dát k dispozici jen velmi malou část funkčnosti platformy .NET. Brzy narazíte na hranice a budete se muset vypořádat se třídami .NET, které tvoří jeho základ. Navíc se se jmenným prostorem My musíte učit další, nadbytečnou syntaxi, neboť dodatečné funkčnosti, které se jinou cestou nenajdou v .NET, nebo jež byste nemohli získat jiným způsobem, jmenný prostor My také neposkytuje.

Proto vám radíme, abyste jmenný prostor My používali jen v těch případech, kdy je to skutečně výhodné. Pokud lze stejné funkčnosti dosáhnout přímo přes třídy .NET, pak byste jim měli dát přednost. Mějte také na paměti, že hledané příklady se na Internetu nenacházejí vždycky ve Visual Basicu 2008. Čtení kódu C# není obtížné, avšak programátoři v C# vždy používají .NET, nikoli jmenný prostor My. Pro profesionální programování a zapracování do .NET je toto nezbytné; jmenný prostor My pomůže jen málo.

Recepty

V 16 kategoriích najdete více než 300 receptů. Stručně zde shrneme, co vás v jednotlivých kategoriích čeká.

Základy

Zde najdete základy, např. pro formátování, bitové operace, pro porovnávání a seřazování objektů a enumerací. Starý příkaz Visual Basicu `Format` dosloužil. Dočtete se, jaké nové, mnohem pohodlnější možnosti pro formátování a pro konverzi čísel a jiných hodnot .NET poskytuje. Manipulace s bity a bajty se v .NET podstatně zjednodušila. Nejenže jazyk Visual Basic 2008 konečně zná operace posuvu, ale platforma .NET nyní nabízí některé třídy pro konverze, jež byly dříve realizovatelné jen velmi obtížně.

Enumerace (výčty) v .NET už nejsou jednoduché pojmenované konstanty, ale také poskytují podporu za běhu, např. pro dotazy na všechny definované hodnoty a jména. Ukážeme vám, jak můžete s enumeracemi zacházet, jak enumerace načítat a vybírat, aniž byste museli pracovat s holými číselnými hodnotami.

Pro řazení seznamů a polí, které obsahují odkazy na objekty, existují všeobecně platné, platformou .NET využívané návrhové vzory (Design Pattern). Vysvětlíme mechanismy, které potřebujete, abyste mohli objekty porovnávat a vyhledávat.

(Téměř) také dosloužily staré funkce `Asc` a `Chr`, neboť pracují pouze se znaky ASCII. Řetězce .NET ale po dobu běhu pracují zásadně s Unicode. Jak převod mezi různými kódováními znaků (ASCII, UTF8, UTF16 atd.) provedete, dozvíte se rovněž v této kategorii.

Datum a čas

Pro práci s datem a časem dává .NET k dispozici rozsáhlé struktury, třídy a metody. Bohužel, existuje zde také několik chyb a pastí, např. při výpočtech v Evropě běžných kalendářních týdnů. S tím vám pomohou naše recepty.

Neustále dotazovaným tématem je výpočet svátků. Na několika receptech ukážeme, jak se spočítají základní data (církevní rok, datum Velikonoc), a dáme vám k dispozici třídu pro oblast se svátky, která se může libovolně rozšiřovat.

Pro historiky je zajímavý přepočítání mezi gregoriánským kalendářem a juliánským datováním.

Aplikace

Po souborech `ini` a po zápisech do registru se dnes konfigurační data ukládají v XML souborech. V kategorii Aplikace představujeme recepty pro čtení dat z konfiguračního souboru dané aplikace. Ukážeme, jak si sami můžete konfigurační soubory založit a ve svém programu je pak číst a zapisovat do nich.

Probereme také nová nastavení aplikací, která Visual Studio 2008 poskytuje. Rovněž tak vysvětlíme, jak je možné v aplikaci centrálně ošetřovat chyby.

Kreslení pomocí GDI+

`GDI+` je grafické API (aplikační programovací rozhraní) nových operačních systémů a od Windows XP výše se instaluje společně s operačním systémem. `GDI` znamená Graphics Device Interface (rozhraní grafických zařízení). Není součástí .NET, avšak všechny grafické úlohy jsou na `GDI+` založeny.

Rozhraní GDI+ je tudíž platformou pro všechny obrázky, které se pomocí programu v .NET vytvoří. Oproti starému, dospělému GDI nabízí celou řadu výhod. Zde jsou některé důležité:

- Systematicky vybudovaný model tříd a funkcí.
- Jednoduše použitelné kreslicí nástroje jako pero, štětec a font.
- Souřadnicové systémy a jejich transformace.
- Průhlednost pomocí alfa-kanálu.

Dnešní grafické karty zpravidla ještě nemají hardwarové urychlovače pro GDI+. V mnoha případech, zejména při použití průhlednosti, proto na rozdíl od GDI může docházet k problémům. Tyto nevýhody však během několika let zmizí, jakmile budou dostupné grafické karty dalších generací.

GDI+ v podstatě nahrazuje všechny funkce GDI, avšak nesmíme zamlčet, že několik možností starého rozhraní GDI už nebude k dispozici, nebo přinejmenším už jen s omezením. Patří sem například možnost, že se opakovaným kreslením XOR předtím nakreslené obrazce opět smažou, aniž by bylo nutné znovu kreslit pozadí.

Tato kniha nemá vysvětlovat základy GDI+. Místo toho na základě několika příkladů ukazuje praktické použití. Protože však kreslicí operace patří mezi nejdůležitější součásti grafického uživatelského rozhraní, věnujeme se jim v knize poněkud zevrubněji. V kategorii Kreslení pomocí GDI+ ukážeme nejprve všelijaké triky a finty s písmem, stínováním a efekty 3D. Monotónní šed' v šedě zobrazených dialogových oknech by poznenáhlu měla patřit minulosti.

Zpracování obrázků pomocí GDI+

Kategorie začíná základy kreslení obrázků, dále v ní najdete recepty pro složité funkce prolínání s ořezáváním a průsvitností, pro zvětšování výřezů obrázků, maximalizaci zobrazení obrázků a pro otáčení a zrcadlení. Ukážeme vám, jak ve svém programu zobrazíte výběr složek s miniaturizovanými obrázky v přehledu a jak můžete generovat náhledy obrázků pro webové stránky.

Kromě toho zde najdete recepty pro zacházení s různými formáty souborů a kodéry. Častá je otázka, jak ukládat obrázky ve formátu JPEG a ovlivnit přitom jejich kvalitu. Také toto zde předvedeme.

Pomocí maticových operací je možné velmi rychle změnit informace o barvách i u větších obrázků. Připravili jsme recepty pro inverzi barevných obrázků nebo pro převod na odstíny šedi. V jednom receptu vysvětlíme testovací program, jehož pomocí dokážete matici barev online změnit a současně vidět účinek.

Pro technická a vědecká vyhodnocení jsou často potřebné filtry pro ostré kreslení. V několika receptech objasňujeme, jak se běžné filtrační algoritmy převádějí do Visual Basicu 2008.

Windows Forms

Možnosti, jež poskytuje .NET pro vytváření oken, jsou téměř nepřehledné. Nově k nim přibyla okna s neobdélníkovými obrysy, jakož i okna zčásti nebo úplně průhledná. Na několika příkladech ukážeme, jak těchto nových efektů můžete využít. Najdete zde recepty k často kladeným otázkám, např. jak se vytvoří a posouvá okno bez titulkové lišty a jak se programem vytvoří snímky obrazovky.

Mnozí programátoři v klasickém VB mají velké problémy při programování dialogových oken pod .NET. Otázky v diskusních skupinách ukazují až příliš často, že dialogy v klasickém VB byly jen zřídka naprogramovány jako objektově orientované, ačkoli již VB5 poskytoval celou řadu možností pro zapouzdření dialogů. Proto považujeme za mimořádně důležité tuto tematiku v několika

receptech zachytit a vysvětlit, jak dialogy pod .NET zapouzdřovat. Třídy dialogů tím získají čistou strukturu, lépe se udržují a mohou být opětovně použity. Děděním můžete navíc dosáhnout toho, že dialogová okna programu budou mít jednotný vzhled.

V souvislosti s dialogovými okny se rovněž často diskutuje o tom, jak se mohou prověřovat uživatelská zadání. Také zde poskytuje .NET některé nové mechanismy.

V této kategorii budou pomocí několika receptů zodpovězeny další, rovněž často kladené otázky, např. jak se může zobrazit úvodní obrazovka (splash screen), nebo jak se metodou Drag&Drop nebo prostřednictvím schránky mohou importovat či exportovat data. Recept pro úvodní obrazovku byl kompletně přepracován a nyní využívá nové možnosti modelu aplikace VB.

Nakonec ukážeme řešení pro banálně znějící formulace úloh, které se při bližším zkoumání jeví jako nanejdůležitější, např. jak se v textovém rámečku zachytí klávesa potvrzení a jak se mohou cesty souborů zkrátit, aby se mohly v ještě čitelné podobě použít v nabídkách.

Windows Controls

Toto je z dobrého důvodu jedna z nejobsáhlejších kategorií. Zacházení s ovládacími prvky se od základu změnilo. Už neexistuje rozdělení na prostředek a kódovou část, jak tomu bylo v klasickém VB. Všechny ovládací prvky se vytvářejí pomocí kódu, v kódu se také inicializují vlastnosti. Od jednoduchých témat, např. jak se mohou nahradit pole ovládacích prvků z klasického VB, až po uživatelsky definované kreslení posuvných seznamů a rozbalovacích seznamů, se budeme intenzivně zabývat programováním vlastních ovládacích prvků. Vysvětlíme, jaké základní třídy jsou pro ty které ovládací prvky vhodné a jaké existují možnosti pro zásah do toku zpráv nějakého ovládacího prvku.

Složitější ovládací prvky pro výběr adresářů a souborů představíme zevrubněji. Mohou se použít přímo pro stavbu vlastních dialogů, pokud ovládací prvky poskytnuté systémem Windows nestačí.

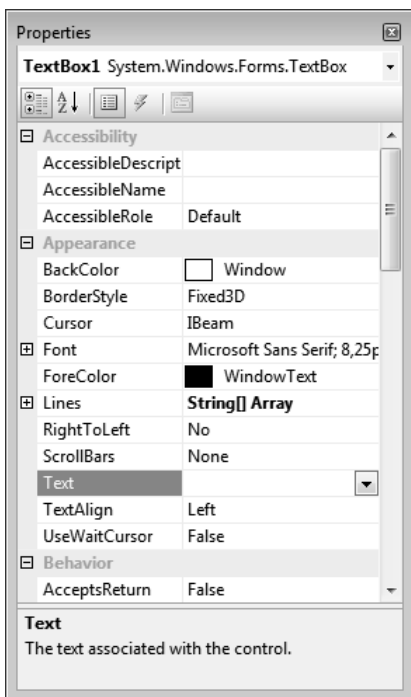
Na příkladu grafického ovládacího prvku pro zobrazení časových segmentů předvedeme, jak se definují ovládací prvky s neobdélníkovým obrysem, jak se transformují souřadnice, jak mohou být události myši zpracovány také v době návrhu, a co je třeba učinit, aby bylo možné nastavit časové segmenty také v okně vlastností.

S víceřádkovými textovými poli zachází .NET poněkud macešsky. V .NET 1.1 byla až dosud v zacházení s víceřádkovými textovými poli velká mezera. Některé informace a nastavení jsou v současnosti dostupné prostřednictvím .NET 2.0, avšak mnohé jiné – stejně jako předtím – jen přes funkce z API. Také k této problematice máme připraveno několik receptů.

Okno vlastností (PropertyGrid)

Jedním z nejnápadnějších ovládacích prvků, které jsou vám k dispozici i pro vlastní aplikace, je ovládací prvek PropertyGrid (okno vlastností). Jedná se o ovládací prvek, kterého využívá návrhář, aby poskytl vývojáři možnost zobrazení a změny vlastností vybraného objektu (ovládacího prvku, okna, nabídky atd.) (viz obrázek 1). Určitě jste ho i vy ve spojení s návrhářem také často použili.

Je lhostejné, zda v návrhářci chcete korektně zobrazit vlastnosti svého vlastního ovládacího prvku, nebo zda chcete v době běhu použít ovládací prvek PropertyGrid ve svých vlastních aplikacích k zobrazení vlastností libovolných objektů: k dispozici máte rozličné možnosti, abyste cíleně ovládali chování okna vlastností. Zejména při profesionálním vývoji ovládacích prvků je nevyhnutelné, abyste se prvkem PropertyGrid zabývali více, neboť je nutné poskytnout uživateli veškerou myslitelnou podporu a zjednodušení pro zacházení s ovládacími prvky.



Obrázek 1: Okno vlastností (PropertyGrid) je jeden z nejvýkonnějších ovládacích prvků a lze ho začlenit i do vlastních aplikací

Vlastně jsme ani nechtěli jednomu jedinému ovládacímu prvku věnovat vlastní kapitolu. Prvek PropertyGrid je však výkonný jako málokterý jiný, často se používá a v dokumentaci MSDN je popsán jen neúplně. Také na Internetu se potřebné informace najdou jen jako velmi roztroušené. Proto jsme pro vás na tomto místě pro okno vlastností vypracovali souvislou kolekci receptů.

Počínaje recepty pro ovládání prvku PropertyGrid, přes atributy pro kategorie, popisy, standardní hodnoty a standardní vlastnosti, vás uvedeme do technik pro zobrazení vlastních editorů, které mohou uživatele při zadávání dat podpořit. Vysvětlíme vám zacházení s výpisy a enumeracemi a jak můžete okno vlastností rozšířit o dodatečná tlačítka odkazů a dodatečné karty.

Ustrnulý údaj zadaných názvů vlastností může ve vícejazyčných programech působit rušivě. Proto zde také naleznete recept, který popisuje, jak můžete zobrazené texty lokalizovat.

Souborový systém

Správa souborového systému patří mezi základní vlastnosti, které musí operační systém podporovat. Bez možnosti vytváření, změn a mazání souborů a adresářů by se všechny informace spolu s vypnutím počítače ztratily. Zůstaly by děrné pásky a karty. V průběhu let se požadavky na souborový systém neustále zvyšovaly.

V kapitole, věnované souborovému systému, najdete recepty právě pro tuto oblast. Protože se však v souborovém systému kromě „normálních“ souborů rovněž ukládá operační systém, upozorňují některé recepty na to, v jakých adresářích uživatel tyto informace uložil.

Jelikož moderní souborový systém je útvarem opravdu dynamickým, nelze se někdy vyhnout hlídání určitých souborů nebo adresářů v programu. Také dnes probíhá výměna dat mezi rozdílnými platformami stále ještě zčásti prostřednictvím souborů. Jeden recept představuje možnosti, jak se mohou v programu hlídat změny souborového systému.

Sít

Bez sítě je už moderní způsob zpracování dat nemyslitelný. Jestliže ještě před krátkým časem byla síť přidavným programem nebo dodatečným ovladačem, který se musel dát operačnímu systému na vědomí, patří dnes funkce sítě k operačnímu systému. Tomu odpovídá i fakt, že mnohé funkce již nerozlišují mezi tím, co je lokální a co globální.

V této části knihy jsme uvedli recepty, které je přesto nutno spíše přiřadit síti, např. jaké síťové adaptéry jsou v počítači zabudovány. Protože IP adresy verze 6 jsou stále známější, jsou zde také dva recepty pro přepočítání starých IP adres na nové a naopak.

Abychom téma webových služeb neponechali celé pouze programátorům v ASP, uvedli jsme na tomto místě také recept k vytvoření webového klienta pro Google ve Visual Basicu 2008.

System

Tento oddíl je z velké části založen na technice, která je poměrně neznámá, přesto však velmi efektivně využitelná: WMI (Windows Management Instrumentation, nástrojové vybavení pro správu Windows). Malý úvod do tematiky jsme zařadili do přílohy.

System lze zhruba rozdělit na dvě oblasti, na hardware a software. Pro hardware zde najdete recepty, které vám umožní zjistit místo na disku nebo číst BIOS. Jsou zde i recepty pro procesor a jeho vlastnosti, nebo pro rozlišení monitoru. Ohledně softwaru zde najdete recepty, pomocí nichž zjistíte, jaký software je na systému instalován, nebo jak můžete pomoci .NET zpracovat registr.

V tomto kontextu nesmějí přirozeně chybět služby a jejich ovládání. Dostaly se sem dva recepty, které mají více společného s komunikací, ale mnohdy najdou v programu uplatnění jako funkčnost systému: odesílání faxů a e-mailů z aplikace. Nový je rovněž recept pro přenos dat pomocí FTP.

Databáze

V této části knihy naleznete hlavně recepty, které mají co činit se správou databáze. Protože existuje dostatek knih k tématu ADO.NET a spíše málo ke správě databáze pomocí vlastního programu, najdete zde recepty, jejichž pomocí dokážete databáze z aplikace nebo samostatným programem ovládat. To je ještě důležitější, neboť s MSDE se žádný odpovídající program Microsoftu nedodává.

V tomto oddílu knihy také najdete recepty pro zabezpečení dat (v kontextu zařízení či mimo něj) a pro automatizaci těchto činností. Dokážete to nejen s lokální databází, nýbrž také s každou databází SQL, pro kterou máte v síti příslušná oprávnění. V receptech předvedeme pochopitelně i cestu opačnou, tj. odstranění zabezpečení.

Některé recepty se zabývají také jinými tématy než jen správou databází. Tak zde najdete recept pro zjištění verze MDAC na počítači, recept jak uložit v databázi obrázky, nebo jak pracovat s Excelem jako s databází.

XML

XML (rozšiřitelný značkovací jazyk) je jedním z nosných pilířů v .NET. Z toho důvodu se také XML používá v mnoha oblastech .NET a je rozličnými způsoby podporován. Základy XML se bohužel příliš detailně zabývat nemůžeme (k tomu účelu existují knihy, které jsou mnohem silnější než tato), ale ukážeme vám nejdůležitější metody pro zacházení s daty a soubory XML.

V této části najdete některé recepty pro základní třídy, jako např. pro `XmlTextWriter` a `XmlTextReader`, nebo pro snímání a psaní obrázků a binárních dat, právě tak jako recepty pro zacházení s objektovým modelem dokumentu (Document Object Model, DOM).

Výrazy jazyka Xpath jsou výkonné konstrukce, které slouží k dotazům v XML. Také o nich zevrubně pojednáme, zejména v souvislosti se jmennými prostory, které u Xpath vyžadují dodatečné činnosti. Poskytneme vám recepty pro třídy `XmlDocument` a `XpathDocument` a vysvětlíme rozdíly.

Mnohé tiskové agentury, vydavatelství časopisů a také MSDN (Microsoft Developer Network, komerční služba Microsoftu) dávají na Internetu k dispozici soubory XML (hesla RSS a RDF) s aktuálními zprávami. Dva recepty ukazují, jak můžete tyto zprávy zpracovat a zobrazit.

V podobě receptů jsme pro vás vypracovali také techniky jako validace souborů XML pomocí schématu XML nebo možnosti pohodlného volání transformací pomocí formátovacích sad (XSLT). Rovněž vysvětlíme zacházení s novým ladicím programem pro XSLT jazyka Visual Basic 2008. Také zde najdete recept, jak vytvořit kód tříd, aniž byste sami museli programovat.

Zajímavým tématem je též paralelní práce se třídami `DataSet` a `XmlDataDocument`, kdy můžete střídavě pracovat s dotazy v jazycích Xpath a SQL.

Na závěr v této kategorii objasníme serializaci libovolných objektů z jazyka XML a do něj. Naleznete recepty pro třídy `XmlSerializer` a `SoapFormatter`.

Vědecké výpočty a křivkové grafy

Fyzikální a technické veličiny se v programování většinou zpracovávají pomocí obyčejných čísel s pohyblivou řádovou čárkou. Typová bezpečnost, která by zabránila zpřeházení například délkových rozměrů a rychlostí, se používá jen zřídkakdy.

Na příkladu některých veličin (délky, plochy atd.) ukážeme, jak můžete cíleně použít objektově orientované programování, jednak abyste zajistili typovou bezpečnost, a za druhé abyste mohli údaje v různých jednotkách automaticky přepočítat na jednu standardní jednotku. Tyto recepty byly kompletně nově zpracovány a nyní intenzivně využívají možností generických tříd. Přidání dalších fyzikálních veličin je pak už hračkou.

K vědeckému zpracování hodnot měření a simulace patří rovněž znázornění v podobě křivkových grafů. Popíšeme, jak určité měřítka os a jak nakreslíte křivky, a zvláště pak, jak můžete naprogramovat spojitý průběh grafů X/Y.

Pokud vám nestačí přesnost, s jakou platforma .NET vypočítává číslo π , pak zde najdete recept, který vám umožní vypočítat π na 2 400 míst.

Úplně nová je série receptů pro lineární algebru. Vysvětlíme, jak lze v dvourozměrných grafech výhodně použít vektorový počet a jak se vypočítají průsečíky různých geometrických tvarů. Zde také najdete recept, jak zjistit, zda se daný bod nachází uvnitř nějakého polygonu.

Zpracování geografických dat

Protože se využití GPS přístrojů značně rozšířilo, rozhodli jsme se při přípravě třetího vydání vytvořit novou kapitolu, věnovanou právě práci s geografickými daty. Do této kapitoly jsme přesunuli několik původních receptů z druhého vydání této knihy a přidali k nim mnoho dalších.

Jak se vyhodnocují data GPS? Jak se vypočítá kurs a vzdálenost k cíli? Předvedeme vám také využití údajů o výškách zemského povrchu, získaných jednou z misí amerického raketoplánu.

DirectX

Kapitola, věnovaná grafice 3D, vám nabízí několik úvodních příkladů. Toto téma by samo zaplnilo celou knihu a zde o něm můžeme pojednat jen v náznacích. Vysvětlíme nezbytné kroky od dotazu na grafickou kartu až po trojrozměrné zobrazení jednoduchých obrazců.

Různé

V poslední kategorii jsme shromáždili všechny recepty, které nejsou dost jednoduché pro kategorii Základy a k tématům ostatních kategorií se nehodí.

Přehrávání zvuku se díky novým funkcím .NET 2.0 zjednodušilo, takže v současné době lze rezignovat na funkce z API. Ukážeme, jak můžete přehrávat zvukové soubory a systémové zvuky, i jak pomocí programu dokážete sami generovat měkce znějící sinusové tóny. Rovněž najdete recepty pro přehrávání not a pro výpočet zvukových kmitočtů.

Za několik receptů nám rovněž stály málo známé ladicí funkce, např. ovládání výstupů pomocí konfiguračního souboru aplikace, výstup ladění a sledování v textových rámečcích, výstup souborů nebo záznamu událostí, nebo vytvoření a využívání vlastních čítačů výkonu (PerformanceCounter). Najdete zde řešení i pro případ, že byste chtěli vědět, jak můžete reagovat na výpis záznamu událostí jiných aplikací.

Dalším receptem jsou rady, jak měřit časy s vysokým rozlišením a na co je přitom třeba dát pozor. Také jak postupovat, chcete-li chybová hlášení funkcí z API zobrazit v nešifrovaném textu.

Typografické konvence

V knize jsou použity následující typografické konvence:

- Klíčová slova Visual Basicu, jakož i označení proměnných, metod, tříd atd. jsou v běžném textu zvýrazněna bezpatkovým neproporcionálním písmem.

Příklad: třída `System.Object` a hodnota `True`

- Výpisy se tisknou následujícím druhem písma. Klíčová slova Visual Basicu jsou vytištěna tučně:

`Dim i As Integer = 123`

- Komentáře ve výpisech jsou napsány kurzívou:

' Toto je komentář

- Jména souborů a adresářů jsou formátována bezpatkovým neproporcionálním písmem:

Příklad: `Text.txt`

- Internetové adresy vypadají takto: *www.addison-wesley.de*

- Texty uživatelských rozhraní, položek menu, popisů tlačítek atd. jsou zapisovány kapitálkami
Příklad: položka nabídky SOUBOR / OTEVŘÍT

Obsah CD u knihy

Na CD u knihy se nachází již zmíněné úložiště, jehož pomocí můžete vytištěné zdrojové kódy vyhledat a prostřednictvím schránky je převzít do svých projektů. Na CD dále najdete všechny příkladové projekty s čísly receptů pro přiřazení receptů k projektům.

Na CD rovněž najdete celou knihu v elektronické podobě.

Errata

Žádná kontrola kvality nedokáže stoprocentně zabránit přehlédnutí chyb. Tak tomu je i u knihy řešení, která má přibližně 1 000 stran. Všechny příklady jsme přirozeně pečlivě vyzkoušeli, avšak nemůžeme vyloučit, že se nám vloudily chyby, nebo že příklad bude ve všech situacích fungovat tak, jak popisujeme.

Pokud tedy najdete chyby nebo nesrovnalosti, nebo budete mít podněty či návrhy na zlepšení, pak nás o tom, prosím, informujte. Naši internetovou adresu, jejíž pomocí s námi můžete navázat kontakt, najdete v předmluvě.

Příloha a odkazy

V příloze jsme pro vás připravili některé důležité informace z pozadí. Ve stručném shrnutí objasňujeme nejdůležitější témata a pojmy ohledně objektově orientovaného programování ve Visual Studiu .NET. Ukazujeme, co se změnilo oproti VB6, blíže zde pojednáváme také o polích a výpisech, rovněž tak o generických typech.

Dalším stěžejním tématem v odkazech je zacházení s Visual Studiem .NET 2008. Nezabýváme se založením projektu (to byste již měli ovládat), ale seznámíme vás s malými, užitečnými detaily, které často zůstávají nevyužity.

Uvádíme tu některé zdroje na Internetu, v nichž můžete získat další informace. Také vysvětlíme, jak můžete nalézt pomoc v probíhajících diskusních skupinách, jak prostřednictvím vyhledávače Google v diskusních skupinách hledat, nebo jak se sami můžete diskusí zúčastnit.

V odkazech se také dotkneme některých základních témat, jež jsme z důvodů nedostatku místa nemohli pojmout do receptů, a stručně je objasníme.

Protože platforma .NET všechno pokrýt nedokáže, jsou v některých receptech použity funkce z API pro Windows. Použité funkce, struktury a konstanty jsme uložili v jedné třídě, která je v odkazech také vytištěna.