



edice Průvodce

Doporučuje časopis

PC WORLD



soubory na
www.gada.cz

Visual Basic .NET 2003

začínáme programovat

Ján Hanák

- Základní pojmy v programování
- Seznámení s jazykem Visual Basic .NET 2003
- Co je to vizuální programování
- Objektově orientované programování
- Vývojová platforma .NET Framework 1.1
- Tvorba dokonalých aplikací .NET



GADA

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**.

Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.





Copyright © Grada Publishing, a.s.



Copyright © Grada Publishing, a.s.

Úvod	9
Struktura knihy	10
Pro koho je kniha určena	12
Co budete potřebovat	13
Typografické konvence	14
Soubory ke stažení	15
Vaše reakce jsou vítané.....	15
Poděkování.....	15



Visual Basic .NET 2003 na první pohled	17
1.1 Základní informace o jazyku Visual Basic .NET 2003	18
1.2 Visual Basic .NET 2003 v počítačovém průmyslu	19
1.3 Vsaďte na Visual Basic .NET 2003.....	21
1.4 Instalace Visual Basicu .NET 2003 Professional.....	23
1.5 První spuštění Visual Studia .NET 2003.....	31
1.6 Vytváříme první projekt	33
1.7 Sémantický rozdíl mezi pojmy projekt a řešení.....	35
1.8 Integrované vývojové prostředí (IDE).....	36
1.9 Charakteristika programování řízeného událostmi	42
1.10 První aplikace v jazyce Visual Basic .NET 2003.....	45
Co nás čeká dále	52



Základy programování v jazyce Visual Basic .NET 2003	53
2.1 Styl programování v jazyce Visual Basic .NET 2003.....	54
2.2 Proměnná, její deklarace, inicializace a použití.....	55
2.3 Obor proměnné, veřejné a soukromé proměnné	57
2.4 Charakteristika a klasifikace datových typů.....	61
2.5 Hodnotové a odkazové datové typy.....	69
2.6 Uživatelsky definované datové typy	71
2.7 Konstanty	73
2.8 Výčtové [enumeriční] typy.....	74
2.9 Nedeklarované proměnné a použití příkazu Option Explicit..	75
2.10 Konverze hodnot datových typů	78
2.10.1 Implicitní konverze.....	78
2.10.2 Použití příkazu Option Strict	80
2.10.3 Explicitní konverze	81

2.11	Procedury Sub a funkce	84
2.11.1	Charakteristika procedur Sub	84
2.11.2	Formální parametry procedury Sub	87
2.11.3	Argument a formální parametr	88
2.11.4	Předávání argumentů hodnotou [ByVal]	89
2.11.5	Předávání argumentů odkazem [ByRef]	90
2.11.6	Předávání argumentů hodnotou a odkazem v praxi	90
2.11.7	Funkce	92
2.12	Operátory	94
2.12.1	Aritmetické operátory	94
2.12.2	Porovnávací operátory	96
2.12.3	Přiřazovací operátory	97
2.12.4	Logické operátory	98
2.12.5	Operátory pro zřetězení	100
2.12.6	Bitové operátory	100
2.12.7	Operátory bitového posunu	103
2.12.8	Speciální operátory	105
2.13	Rozhodovací konstrukce a řízení běhu programu	105
2.13.1	Rozhodovací konstrukce If-Then a její varianty	105
2.13.2	Rozhodovací konstrukce Select-Case	110
2.14	Cykly	112
2.14.1	Cyklus For-Next	113
2.14.2	Cyklus For Each-Next	115
2.14.3	Cyklus Do While-Loop	116
2.14.4	Cyklus Do-Loop While	118
2.14.5	Cyklus Do Until-Loop	119
2.14.6	Cyklus Do-Loop Until	119
2.14.7	Cyklus Do-Loop	120
2.15	Pole	120
2.16	Softwarové aplikace a chyby	124
2.16.1	Chybové výjimky, jejich zachytávání a ošetřování chyb	124
2.16.2	Nestrukturovaná a strukturovaná správa chyb	125
2.17	Jmenné prostory	126
	Co nás čeká dále	127



Vizuální programování 129

3.1	Vizuální programování a Visual Basic .NET 2003	130
3.2	Formulář pod drobnohledem	130
3.2.1	Přehled důležitých událostí formuláře	133
3.2.2	Vytváření zpracovatelů událostí formuláře	135
3.2.3	Modální a nemoďální formuláře	136
3.2.4	Experimenty s formuláři	138
3.2.5	Vytváření formulářů pomocí programového kódu	138

3.2.6 Sestrojení průhledného formuláře	139
3.2.7 Vystředění formuláře na obrazovce počítače ...	140
3.2.8 Implementace vizuálních stylů systému Windows XP	140
3.3 Budování grafického rozhraní aplikací pomocí instancí ovládacích prvků a komponent	142
3.4 Zásady správného návrhu vizuálního rozhraní aplikací	143
3.5 Přehled ovládacích prvků a komponent	144
Co nás čeká dále	149



Úvod do objektově orientovaného programování 151

4.1 Základní myšlenka OOP	152
4.2 Principy OOP	153
4.2.1 První pilíř OOP: abstrakce	154
4.2.2 Druhý pilíř OOP: zapouzdření	154
4.2.3 Třetí princip OOP: skrývání dat	154
4.2.4 Čtvrtý princip OOP: dědičnost	155
4.2.5 Pátý princip OOP: polymorfismus	156
4.2.6 Šestý princip OOP: další použitelnost programového kódu	157
4.3 Vytváříme první třídu v jazyce Visual Basic .NET 2003...	157
4.4 Aplikace dědičnosti při tvorbě odvozených tříd	160
Co nás čeká dále	161



.NET Framework 1.1 jako prostředí pro vývoj aplikací .NET 163

5.1 Vývojová platforma .NET Framework 1.1	164
5.2 Architektura vývojové platformy .NET Framework	164
5.2.1 Společné běhové prostředí	165
5.2.2 Assembly a její struktura	167
5.2.3 Bázová knihovna tříd	167
5.2.4 Společný typový systém	168
5.2.5 Společná jazyková specifikace	169
5.3 Automatická správa paměti pomocí softwarové služby Garbage Collection	169

Závěr 171

Rejstřík 173



Úvod

Úvod

Vážení čtenáři,

do rukou se vám dostává publikace věnovaná prozatím nejnovější verzi programovacího jazyka *Visual Basic .NET* s označením 2003. *Visual Basic .NET 2003* je v současnosti snad nejpopulárnějším programovacím nástrojem pro vývoj počítačových aplikací, určených pro platformu *.NET Framework 1.1*. Pohled do historie však říká, že cesta *Visual Basicu* k dokonalosti nebyla vůbec jednoduchá ani přímočará.

Dějiny tohoto programovacího jazyka jsou bezesporu velmi zajímavé. Vždyť to byl právě *Visual Basic*, který přinesl koncepci vizuálního programování a právem byl ve svém názvu hrdý na slovíčko „Visual“. Od uvedení první verze *Visual Basicu* v roce 1991 se tento jazyk, podobně jako jeho další kolegové, pozvolna vyvíjel tak, aby se mohl stát plnohodnotným vývojovým nástrojem pro platformu *Windows* společnosti *Microsoft*. Jednou z přelomových verzí byla „čtyřka“, uvedená na softwarový trh v roce 1995. *Visual Basic 4.0* byl zaměřen na tehdejší nejmodernější platformu, kterou se stal operační systém *Windows 95*. Ambice *Visual Basicu* byly vysoké, bylo totiž zapotřebí demonstrovat sílu tohoto nástroje při vývoji aplikací založených na pokročilém grafickém uživatelském rozhraní systému *Windows 95*. Je třeba konstatovat, že *Visual Basic 4.0* tuto bitvu vyhrál, a to zejména díky prvotní implementaci základních objektově orientovaných prvků a pokročilým technologiím, mezi něž patřilo OLE a automatizace aplikací. Další verze *Visual Basicu* (5.0 a 6.0) přinesly kromě nespočetných vylepšení také komplexní vývojové prostředí pro rychlý vývoj aplikací (jde o tzv. prostředí **RAD – Rapid Application Development**), v němž byla radost pracovat. Zatímco verze 5.0 a 6.0 *Visual Basicu* byly uvedeny relativně rychle po sobě (v letech 1997 a 1998), po vydání šesté verze nastalo jakési pomyslné ticho před bouří, které věštilo, že vývojáři v Redmondu pracují tentokrát na něčem opravdu velkém.

Když se v roce 2002 objevila první verze *Visual Basicu .NET*, která pracovala pod křídly platformy *.NET Framework 1.0*, začala se psát nová kapitola historie tohoto programovacího jazyka. Mnozí programátoři, odchovaní na *Visual Basicu*, byli zprvu zcela ohromeni obrovskou silou nové verze a jejím programovacím aparátem. *Visual Basic .NET* přinesl kompletní začlenění koncepce objektově orientovaného programování, včetně dědičnosti, práce s konstruktory a finalizéry, přetěžování metod a jiných účinných programovacích postupů. V polovině roku 2003 společnost *Microsoft* uvedla na trh modernizovanou verzi *Visual Basicu .NET* s přídomkem 2003. Kromě *Visual Basicu .NET 2003* byl představen i nový vývojový rámec *.NET Framework* s označením 1.1. Publikace, kterou právě držíte v rukou, se zabývá *Visual Basicem .NET 2003*, tedy zatím poslední dostupnou verzí tohoto oblíbeného programovacího jazyka. Cílem publikace je naučit vás pracovat v prostředí *Visual Basicu .NET 2003* tak, abyste byli schopni vyvíjet své první plně kvalifikované aplikace pro platformu *.NET Framework 1.1*.

Struktura knihy

Publikace *Visual Basic .NET 2003 – začínáme programovat* je rozdělena do několika samostatných kapitol:

1. *Visual Basic .NET 2003* na první pohled.
2. Základy programování v jazyce *Visual Basic .NET 2003*.
3. Vizuální programování.

4. Úvod do objektově orientovaného programování.
5. *.NET Framework 1.1* jako prostředí pro vývoj aplikací .NET.

V první kapitole nazvané *Visual Basic .NET 2003 na první pohled* se dozvíte, jakou roli hraje *Visual Basic .NET 2003* v současném světě vývoje softwarových aplikací. Zjistíte, co vám může nabídnout a proč je vhodné si vybrat pro vývoj aplikací na platformě *.NET Framework 1.1* právě tento programovací nástroj. Seznámíte se s vestavěným integrovaným vývojovým prostředím (**IDE – Integrated Development Environment**) a dozvíte se, jaká je generická podoba procesu vývoje aplikací pomocí *Visual Basicu .NET 2003*. Finálním bodem první kapitoly je pak sestavení první standardní aplikace pro operační systém *Windows*, která bude vyhovovat kritériím vývojové platformy *.NET Framework*.

Zaměření druhé kapitoly výstižně charakterizuje již její název: *Základy programování v jazyce Visual Basic .NET 2003*. Ačkoliv *Visual Basic .NET 2003* je nástroj pro vizuální vývoj aplikací, přesto je zapotřebí si v jistém okamžiku vyhrnout rukávy, otevřít editor pro zápis programového kódu a pustit se do psaní programových instrukcí. Pokud s programováním teprve začínáte, možná máte jisté obavy, zdali nebude psaní programového kódu přece jenom příliš náročné. Podobné obavy však můžete hodit směle za hlavu, a to z několika důvodů:

1. *Visual Basic .NET 2003* je programátorsky přívětivý, což znamená, že jednotlivé programovací konstrukce jsou sestavovány logicky. Samotný zdrojový kód programu je ve většině případů srozumitelnější a snáze pochopitelný než u jiných programovacích jazyků.
2. Integrované vývojové prostředí *Visual Basicu .NET 2003* obsahuje několik nástrojů, které vám s psaním programových příkazů ve velké míře pomohou. Jde především o technologii **IntelliSense**, která sleduje vaši práci s kódem a nabízí názvy programových prvků, které můžete potřebovat.
3. Opravdu skvělým pomocníkem je i nový styl nápovědy, který byl do *Visual Basicu .NET 2003* nově zabudován. Jedná se o tzv. dynamickou nápovědu, která monitoruje vaše kroky a připravuje ta témata nápovědy, která byste mohli potřebovat nebo která by pro vás mohla být zajímavá. Není tedy již nutné, abyste v nouzi vyhledávali požadované informace, protože moderní vyhledávací nástroje dynamické nápovědy vám je doručí okamžitě a zcela automaticky.
4. *Visual Basic .NET 2003* je schopný odhalit množství chyb již v okamžiku, kdy zadáváte programový kód do editoru. Tak se výrazně snižuje množství práce, které je nutné vynaložit pro zjištění potenciálních problémových oblastí.
5. Veškeré postupy, na něž v této publikaci narazíte, jsou popsány stylem „krok za krokem“. Z vlastní zkušenosti vím, že tento styl výkladu je mezi začínajícími programátory velice oblíbený. Abyste probíranou látku pochopili co možná nejrychleji, jsou pro vás připraveny názorné ilustrace, vývojové diagramy a schémata, která v grafické podobě doplňují probíranou problematiku.

Jak sami uvidíte, programování v jazyce *Visual Basic .NET 2003* je zábavné a vzrušující zároven. Druhá kapitola je zaměřená na představení základních programovacích entit, s nimiž začínající programátoři zcela jistě přijdou do styku. Dozvíte se, co jsou proměnné, jak se deklarují a inicializují. Nahlédnete do problematiky datových typů proměnných a seznámíte se s implicitními a explicitními konverzními mechanismy, pomocí nichž lze měnit charakter hodnot proměnných. Poznáte také konstanty, výčtové typy (enumerace), metody, funkce, argumenty, formální parametry, cykly, rozhodovací konstrukce a mnoho dalšího. Sečteno a podtrženo, po absolvování základů programování ve *Visual Basicu .NET 2003* budete rozu-

mět všem podstatným programátorským pojmům a budete schopni vysvětlit práci napsaného programového kódu.

Pokud se podíváme do historie programování pro platformu *Windows*, zjistíme, že to byl právě *Visual Basic*, který přinesl opravdové vizuální programování. Vizuální programování zásadním způsobem ovlivnilo dosavadní styl vývoje počítačových aplikací. Ve *Visual Basicu* bylo možné velice rychle vytvářet grafické uživatelské rozhraní, jehož prostřednictvím aplikace komunikovaly se svými uživateli. Touto výtečnou vlastností disponuje samozřejmě i *Visual Basic .NET 2003*. Ovšem předchozí tvrzení není zcela správné – *Visual Basic .NET 2003* ve skutečnosti otevírá bránu do nového vizuálního světa vývoje aplikací a současně poskytuje všechny nástroje, jejichž pomocí vytvoříte dosud nevidané aplikace.

Jelikož všudypřítomný trend vizuálního programování nelze v žádném případě přehlížet, věnujeme mu celou kapitolu s názvem *Vizuální programování*. Naším cílem bude zejména vývoj standardních aplikací pro *Windows*, a proto se seznámíme s objektovou knihovnou **Windows Forms**, která tvoří pomyslný základní kámen vývoje aplikací založených na formulářích *Windows*. Jistě víte, že grafické uživatelské rozhraní dnešních aplikací je tvořeno širokým spektrem ovládacích prvků, mezi něž patří například ukazatel průběhu (**ProgressBar**), textové pole (**TextBox**), či tlačítko (**Button**). I když je množina dostupných ovládacích prvků bohatá, nesusoudíme se na úplný popis všech ovládacích prvků. Místo toho se budeme koncentrovat jenom na nejběžněji používané ovládací prvky, které můžete nalézt v takřka každé aplikaci. Rovněž nezapomene ani na zásady správné tvorby grafického rozhraní aplikací a na speciální tipy, které vám pomohou aplikace „vyšperkovat“ (například se naučíte, jak aplikaci přimět, aby dovedla využívat nové vizuální styly operačního systému *Windows XP*).

Již několik let vládne ve světě programování jeden jediný programovací styl, známý jako objektově orientované programování, zkráceně OOP. Podobně jako celá platforma *.NET Framework 1.1* je i *Visual Basic .NET 2003* plně objektově orientovaný. První exkurzi do OOP najdete ve čtvrté kapitole s názvem *Úvod do objektově orientovaného programování (OOP)*. A co se zde dozvíte? V první řadě zjistíte, co ve skutečnosti pod poněkud tajemným názvem OOP rozumíme a jaká je hlavní koncepce tohoto programovacího stylu. Proniknete do tajů takových pojmů, jako abstrakce, zapouzdření, skrývání dat, dědičnost a polymorfismus. Společně s představením základního terminologického aparátu bude ozřejměn také vztah mezi třídou, jakožto primární entitou OOP, a instancemi třídy, neboli objekty. Řeč bude i o tom, jak vám může OOP pomoci při opětovném použití již vytvořených fragmentů programového kódu.

Zájemce o hlubší studium problematiky platformy *.NET Framework 1.1* zcela jistě potěší poslední kapitola s názvem *.NET Framework 1.1 jako prostředí pro vývoj aplikací .NET*. Domnívám se, že i pro programátory-začátečníky je důležité vědět, z čeho se skládá a jak pracuje vývojové prostředí, v němž jejich aplikace běží. Proto vám vřele doporučuji prostudovat výklad, který je věnován báze knihovně tříd **.NET Framework 1.1 Class Library**, společnému běhovému prostředí (**Common Language Runtime**), společnému typovému systému (**Common Type System**) a společné jazykové specifikaci (**Common Language Specification**). Nicméně ujít byste si neměli nechat ani proces sestavení základní jednotky aplikace (**assembly**), **Just-In-Time (JIT)** kompilaci aplikačního kódu a samotný běh aplikace. Blíže bude charakterizována také problematika hodnotových a odkazových (referenčních) datových typů. Závěr bude patřit podrobnějšímu pohledu na automatickou správu paměti, kterou zabezpečuje softwarová služba **Garbage Collection**.

Pro koho je kniha určena

Kniha *Visual Basic .NET 2003 – začínáme programovat* je vhodná pro všechny čtenáře, kteří se chtějí naučit rychle a efektivně programovat v moderním programovacím jazyce *Visual*

Basic .NET 2003. Je přitom zcela nepodstatné, zdali jste se již někdy s programováním setkali či nikoliv. Tato publikace si nese „všechno s sebou“, což znamená, že všechny důležité informace získáte na jediném místě, a nebudete je tedy muset čerpat z dalších zdrojů. Výklad je koncipován tak, abyste si nové poznatky a vědomosti osvojovali postupně a v logickém sledu. Přesto se ovšem předpokládá, že umíte velice dobře pracovat s operačním systémem *Microsoft Windows*. Nemusíte sice disponovat hlubšími znalostmi o architektuře systému, ovšem obstojná práce s grafickým uživatelským rozhraním by vám neměla činit žádné potíže.

Pokud patříte mezi úplně nováčky, kteří s programováním teprve začínají, bude pro vás nejlepší, když začnete úvodní kapitolou *Visual Basic .NET 2003 na první pohled* a budete pokračovat postupně, tedy kapitolu za kapitolou. Podobné doporučení platí také pro programátory, kteří přicházejí z jiných programovacích prostředí, jako například *Visual Basic 6.0* či *Visual C++ 6.0*. Na druhou stranu pokud pracujete s jinými programovacími jazyky platformy *.NET Framework 1.1* (například s *Visual C# .NET 2003*), budete se patrně soustředit především na ovládnutí jiného stylu psaní programového kódu, k čemuž vám pomůže zejména druhá kapitola *Základy programování ve Visual Basicu .NET 2003*. Kniha bude rovněž užitečná pro vývojáře, programující v nižší verzi *Visual Basicu .NET (2002)* na platformě *.NET Framework 1.0*. Verze *2002* a *2003 Visual Basicu .NET* se v některých koncepcích a prvcích liší, například pomocí *Visual Basicu .NET 2003* mohou programátoři přímo vytvářet aplikace pro počítače do dlaně, PDA a jiná mobilní zařízení. Jestliže byste se rádi blíže seznámili s platformou *.NET Framework 1.1*, nalistujte asi nejdříve pátou kapitolu *.NET Framework 1.1 jako prostředí pro vývoj aplikací .NET*.

Co budete potřebovat

Abyste mohli plně využít všechny poznatky a aplikovat postupy a programové algoritmy, které naleznete v této publikaci, budete potřebovat následující artefakty:

- Počítač s procesorem Intel Pentium III, taktovaným na 600 MHz (nebo lepší), 256 MB operační paměti, přibližně 5 GB volného místa na pevném disku, optickou mechaniku CD-ROM nebo DVD-ROM a sedmnáctipalcový monitor s rozlišením 1024 × 768 obrazových bodů při 16bitové barevné hloubce.
- Operační systém *Windows XP Professional* nebo *Windows XP Home Edition*, případně *Windows 2000 Professional*.
- *Visual Basic .NET 2003 Professional*.¹
- Připojení k internetu o rychlosti minimálně 56 kB/s, abyste mohli stahovat nové aktualizace *Visual Basicu .NET 2003* a další informace, které se k tomuto softwarovému produktu vážou.
- Jelikož vývojové prostředí *Visual Basicu .NET 2003* není lokalizováno do českého jazyka, je nutné, abyste ovládali počítačovou angličtinu, a to alespoň na mírně pokročilé úrovni. Všeobecně lze říci, že čím lépe tento jazyk ovládáte, tím bude pro vás práce s *Visual Basicem .NET 2003* i s jeho doprovodnou dokumentací příjemnější.

¹ *Visual Basic .NET 2003 Professional* není dostupný v samostatné edici, nýbrž je součástí *Visual Studio .NET 2003 Professional*. Pokud nemáte přístup k této verzi *Visual Basicu .NET 2003*, můžete použít také *Visual Basic .NET 2003 Standard*, který je sice funkčně méně vyspělý, ovšem za výhodnou cenu si jej můžete opatřit v samostatném balení. Pokud se však v budoucnosti hodláte věnovat pokročilejšímu studiu *Visual Basicu .NET 2003* a také dalších programovacích jazyků *Visual Studio .NET 2003 (Visual C# .NET 2003, Visual C++ .NET 2003, Visual J# .NET 2003)*, vřele vám doporučuji, abyste si raději zakoupili kompletní softwarový produkt *Visual Studio .NET 2003 Professional*.

- V neposlední řadě budete potřebovat také notnou dávku energie, chuti a času, které budete ochotni investovat do studia moderního programovacího jazyka. V každém případě si však můžete být jisti, že tato investice se vám v budoucnosti vrátí i s pomyslnými úroky. Společnost *Microsoft* totiž vkládá do platformy *.NET Framework* a do *Visual Basicu .NET* velké naděje, a proto je více než jisté, že odborníci na tuto oblast budou na trhu práce patřit k těm nejžádanějším.

Typografické konvence

Při čtení publikace se můžete setkat s některými typografickými prvky, které byly do textu implementovány v zájmu zvýšení přehlednosti a uživatelské přívětivosti.

Veškerý text, který neoznačuje programový kód ani názvy žádných jiných programových entit, je psán standardním stylem písma.

Pro označení názvů položek nabídek, typů projektů a dialogových oken či internetových adres bude použito **tučné písmo**. Pro komerční názvy softwarových produktu bude použita *kurziva*. Klávesy a klávesové zkratky budou označeny KAPITÁLKAMI.

Názvy programových elementů, jako jsou proměnné, funkce, formální parametry či instance tříd, o nichž je řeč ve výkladu, jsou psány *neproporcionálním písmem*. (Příklad: „Proměnná *PočetHodůKostkou* byla implicitně inicializována nulovou hodnotou, avšak jestliže chceme tuto proměnnou použít v metodě *HoditKostkou*, musíme do ní nejprve uložit nějakou smysluplnou hodnotu.“)

Výpisy programového kódu jsou formátovány pomocí stejného písma, *Courier New*. Toto písmo je ve většině editorů pro zápis zdrojového kódu programu nastaveno jako standardní, přičemž podobná je situace i u *Visual Basicu .NET 2003*. V uváděných fragmentech zdrojového kódu se budou nacházet i komentáře (ty snadno identifikujete pomocí apostrofu, jenž se nachází před každým řádkem s komentářem). Komentáře jsou, podobně jako zdrojový kód programu, zapsány prostřednictvím *neproporcionálního písma*. Příklad:

```
Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click
    'Komentář: Zde je deklarována proměnná.
    Dim Proměnná As System.Int32
End Sub
```

V textu publikace se vyskytují také odstavce označené ikonou, které si můžete představit jako ostrůvky, nabízející další doplňující informace, případně upozorňující na některá důležitá fakta, která byste měli mít v každém případě na paměti.



Upozornění sděluje uživateli důležité informace, které se vztahují k probíranému tématu a které nelze v žádném případě přehlížet. Když tuto značku spatříte v textu, rozhodně jí věnujte pozornost.



Poznámka obsahuje dodatečné informace, o nichž byste měli vědět. Ačkoliv důležitost údajů, které tento odstavec nabízí, není tak vysoká jako u upozornění, přesto je vhodné, abyste se s nimi seznámili.



Tip zase naznačuje alternativní postup či algoritmus řešení zkoumaného problému, který by mohl zrychlit nebo zefektivnit vaši práci. Může jít o použití klávesové zkratky nebo také o tip, jenž urychlí provádění zdrojového kódu.

+

Soubory ke stažení

Tato publikace obsahuje mnoho ukázek programového kódu, na nichž jsou ilustrovány základní programátorské koncepce a postupy. Dobrou zprávou je, že všechny důležité programové ukázky jsou pro vás volně dostupné na internetu (na adrese www.grada.cz), odkud si je můžete bezplatně stáhnout. To znamená, že budete-li chtít jistý zdrojový kód vyzkoušet, nemusíte zbytečně ztrácet čas přepisováním programové ukázky řádek po řádku z knihy. Místo toho si ve *Visual Basicu .NET 2003* otevřete příslušný projekt a můžete okamžitě začít s testováním. Všechny dostupné ukázky jsou vhodně uspořádány a veškerý zdrojový kód je náležitě okomentován, takže vše je připraveno pro to, aby bylo vaše studium programovacího jazyka *Visual Basic .NET 2003* rychlé a efektivní.

Vaše reakce jsou vítané

Autor této publikace vám bude velice vděčný, když mu sdělíte, jak se vám tato kniha líbí. Je k zastížení na adrese elektronické pošty hanja@stonline.sk, což je místo, kam můžete směřovat své názory, dotazy či připomínky.

Poděkování

Na tomto místě bych chtěl vyjádřit své srdečné díky všem lidem z nakladatelství Grada Publishing, kteří mi umožnili využít mé programátorské znalosti a připravit tuto publikaci. Především bych velice rád poděkoval zejména panu Mgr. Jaroslavu Škárkovi za jeho velice vstřícný přístup a snahu, která ve značné míře přispěla k vydání této knihy. Mé poděkování a uznání však patří i dalším pracovníkům, kteří vynaložili své úsilí, aby se tato publikace mohla dostat až do rukou čtenářů. Takže ještě jednou vřelé díky!



Visual Basic .NET 2003 na první pohled

1. Visual Basic .NET 2003 na první pohled

V této kapitole se dozvíte:

- Jakou roli hraje *Visual Basic .NET 2003* v současném světě vývoje softwarových aplikací.
- Co vám může *Visual Basic .NET 2003* nabídnout a proč si vybrat pro vývoj aplikací pracujících na platformě *.NET Framework 1.1* právě tento programovací jazyk.
- Co obsahuje krabicová verze *Visual Basicu .NET 2003* a jak probíhá instalace tohoto softwarového produktu.
- Co je integrované vývojové prostředí (IDE) a z čeho se skládá.
- Jak sestavit pomocí *Visual Basicu .NET 2003* vaši první standardní aplikaci pro *Windows*, která plně vyhovuje požadavkům platformy *.NET*.

1.1 Základní informace o jazyku Visual Basic .NET 2003

*Visual Basic .NET 2003*¹ je moderní, objektově orientovaný programovací jazyk, jenž byl navržen pro vytváření široké škály aplikací pro běh na lokálních i vzdálených počítačových stanicích. Je přitom zcela nepodstatné, zda se chystáte vydat na dráhu programátora aplikací pro operační systém *Windows*, nebo pro web – při použití *Visual Basicu* si můžete být jisti, že produktivita vaší práce poroste do závratných výšin. Po několikaletých zkušenostech s tímto programovacím jazykem (i s jeho předchozími verzemi) mohu potvrdit, že na to, abyste vyřešili určitý programový úkol, je ve *Visual Basicu* nutné vynaložit menší množství programátorské práce než v jiných programovacích jazycích.

Visual Basic mě poprvé okouznil ve svém čtvrtém vydání (uvedeném v roce 1995). Přestože jsem programoval také v dalších jazycích (*Pascal*, *C* a *C++*), *Visual Basic* měl už tehdy v sobě cosi originálního a neopakovatelného. Snad to byl nový styl vývoje aplikací, rychlá práce s programovým kódem, účinná automatizace aplikací nebo skutečné vizuální programování. V dřívějších verzích byla role *Visual Basicu* mnohdy neprávem zamlžována, zejména ze strany programátorů, kteří podle svého názoru používali „skutečné“ programovací jazyky jako *C* a *C++*. Těmto výročkám jsem nikdy nepřikládal žádnou váhu, a to zejména proto, že s uvedenými jazyky jsem měl osobní zkušenosti (nevím však, zda autoři podobných tvrzení někdy skutečně pracovali s *Visual Basicem* – nedivil bych se, kdyby nikoliv).

Příchod *.NET* verze *Visual Basicu* ovšem všechny pochyby rozptýlil. Stal se z něho plnohodnotný programovací jazyk, bezpečně zakotvený na objektově orientovaných základech. *Visual Basic .NET* však nebyl osamocěným jezdcem, společně s ním spatřily světlo světa také další programovací nástroje, a to *Visual C# .NET* a *Visual C++ .NET*. Všechny programovací jazyky byly podepřeny všeobjímající vývojovou a exekuční platformou s názvem *.NET Framework* verze 1.0. Právě nová platforma pro vývoj a běh aplikací *.NET* znamenala vskutku revoluční přínos do poněkud stojatých vod programování aplikací. Přinesla totiž dlouho požadovaný standard a zcela novou architekturu programových elementů, která v notné míře ovlivňuje proces sestavování a programování aplikací, tedy něco, co nepředpokládali ani ti nejlepší softwaroví kouzelníci. Vytvořené aplikace už totiž nejsou předkládány přímo do strojového

¹ Tato publikace se primárně soustřeďuje na popis programovacího jazyka *Visual Basic .NET 2003*. Jelikož je tento název poněkud dlouhý, další text bude pracovat také se zkráceným ekvivalentem *Visual Basic .NET*. Pokud tedy nebude výslovně napsáno jinak, bude slovní spojení *Visual Basic .NET* nahrazovat plně kvalifikovaný název softwarového produktu *Visual Basic .NET 2003*.

kódu, kterému rozumí instrukční sada procesoru. Místo toho je všechny aplikační kód převeden do podoby speciálního nízkourovňového programovacího jazyka s názvem *Microsoft Intermediate Language* (zkráceně *MSIL* nebo také *IL*). Dojde-li ke spuštění aplikace, je startovní kód aplikace nejprve zaveden do paměti počítače, kde posléze dojde k přeložení potřebných instrukcí z *MSIL* kódu do nativního (strojového) kódu. Přeložení *MSIL* kódu je realizováno „na požádání“ pomocí vestavěného Just-In-Time (JIT) kompilátoru. Takto upravený kód se v dalším kroku stává soustem pro procesor a jeho instrukční sadu.

Velkým přínosem byla bezpochyby skutečnost, že výsledným produktem kompilátorů všech zúčastněných programovacích jazyků (*Visual Basic .NET*, *Visual C# .NET* a *Visual C++ .NET* s *Managed Extensions*) se stal právě *MSIL* kód. Tak se rázem setřely rozdíly mezi startovní rychlostí aplikací napsaných pomocí různých programovacích jazyků, protože ať pracujete v jakémkoliv z nich, finální kód vytvořené aplikace bude stejně tvořen programovými instrukcemi jazyka *MSIL*.

Platforma *.NET Framework 1.0* obsahovala nepřeberné množství novinek a inovativních koncepcí, jejichž cíl byl jediný – připravit takové prostředí, které by na jedné straně maximalizovalo produktivitu a rychlost práce programátorů, na straně druhé by jim zase nabízelo příležitost sestavit a odladit i velice sofistikované počítačové programy.

V první polovině roku 2003 se na trh dostala nová verze platformy *.NET Framework*, označovaná jako *1.1*. Další reinkarnace se dočkal také *Visual Basic .NET*, v jehož názvu svítí číslovka 2003. Mnozí programátoři říkají, že jde zatím o nejpovedenější verzi *Visual Basicu* a v tomto směru s nimi plně souhlasím.

Čtete-li tuto knihu, je pravděpodobné, že jste se rozhodli vstoupit do světa *Visual Basicu*. Snad vás potěším, když prohlásím, že na seznámení s *Visual Basicem* snad nikdy nebyla vhodnější doba než právě teď!

1.2 Visual Basic .NET 2003 v počítačovém průmyslu

Pokud se zeptáte různých programátorů na jejich oblíbený programovací jazyk, je pravděpodobné, že dostanete úplně odlišné odpovědi. Tato skutečnost je způsobena tím, že existuje poměrně velké množství programovacích jazyků, které se odlišují v mnoha attributech, například syntaxí programového kódu, počtem vestavěných programovacích konstrukcí či ochotou spolupracovat s aplikacemi vytvořenými v jiných programovacích jazycích. Ve světě programování se můžete setkat s jazyky několika úrovní. Než si představíme reprezentační vzorek programovacích jazyků, které se v současné době teší značné oblibě, měli bychom si vysvětlit termín „úroveň programovacího jazyka“.

Zjednodušeně lze říct, že úroveň programovacího jazyka vyjadřuje míru programové abstrakce, kterou nabízí daný programovací jazyk svému uživateli. Vycházejme ze všeobecné teze, podle níž software existuje proto, aby ovládal technické součástky počítače, tedy hardware. To je základní úloha, kterou musí každý software zvládat. Samotné řízení hardwaru však může být realizováno na několika úrovních, podle toho, jakou abstrakci hardwaru dokáže programovací jazyk zprostředkovat.

Pomocí jazyka symbolických instrukcí² můžeme ovládat hardwarové komponenty explicitně a velice rychle. Na druhou stranu kód zapsaný v tomto jazyce bývá často velmi složitý, obtížně se čte a obtížně se v něm hledají programové chyby. Proto říkáme, že jazyk symbolických instrukcí je jazykem nízké úrovně – jeho míra programové abstrakce od hardwaru počítače

² Jazyk symbolických instrukcí je někdy nazýván také assembler.

je velmi malá. Přesto jsou některé aplikace napsány právě v tomto jazyce. Důvodem je již zmíněná rychlost provádění zdrojového kódu programu, která je u jistých aplikací, jako jsou třeba počítačové hry, prvotná.

Přestože mnozí profesionální programátoři si s jazykem symbolických instrukcí dobře rozumí, většina vývojářů potřebuje při vytváření aplikací větší komfort a programátorskou přívětivost. Tyto požadavky kdysi plně uspokojily programovací jazyky střední úrovně, přesněji jazyky C a C++. Již samotná syntaktická struktura obou jazyků se znatelně lišila od jazyka symbolických instrukcí. Jazyk C uvedl přátelštější programové schéma, které dovolovalo snadnější a přehlednější tvorbu aplikací. Podobně byl zaměřen i programovací jazyk C++, jenž dovedností svého staršího bratříčka rozšířil o plnou podporu objektově orientovaného programování. Programovací jazyky střední úrovně přinášejí vyšší stupeň programové abstrakce – není již nutné psát symbolické instrukce, protože jednotlivé programové entity jsou vhodně pojmenované tak, aby práce s nimi byla intuitivnější. Na druhé straně jazyky střední úrovně dokáží také zprostředkovat přímý přístup k hardwaru počítače (je například možné získat plný přístup k vyhrazené počítačové paměti nepřímo pomocí ukazatelů, aplikovat zásady ukazatelové aritmetiky či ovládat počítačové porty).

Programovací jazyky C a C++ používá i dnes mnoho programátorů, ovšem ne v původní podobě, nýbrž v některém z inovovaných vizuálních vydání. K těm patří zejména úspěšný produkt společnosti *Microsoft* s názvem *Visual C++ 6.0*. Fenomén vizuálního programování zažil boom zejména ve druhé polovině devadesátých let minulého století. Kromě *Visual C++ 6.0* byl představen samozřejmě také *Visual Basic 6.0*. Tyto programovací jazyky představovaly první generaci skutečných vizuálních nástrojů a záhy se proto staly vlajkovými loděmi společnosti *Microsoft* pro vizuální vývoj počítačových aplikací. Zaměření na vizuální programování se projevilo s odstupem času jako velice chytrý strategický krok, jenž byl završen uvedením platformy *.NET Framework 1.1* a *Visual Studio .NET 2003* v roce 2003. Při *.NET* programovacích jazycích mluvíme o druhé generaci nástrojů pro vizuální programování.

Všechny programovací jazyky obsahující ve svém názvu magickou formulku *Visual*, případně také *.NET*, se řadí do kategorie vyšších programovacích jazyků. Pro vyšší programovací jazyky jsou charakteristické následující vlastnosti:

1. Vysoká úroveň abstraktní programové vrstvy, která odděluje programátory od hardwarové báze. Programátor se tak vůbec nemusí zatěžovat prováděním jistých operací nízké či střední úrovně, protože ty za něj vyřeší samotný programovací jazyk, respektive jedna z implementovaných podpůrných jazykových vrstev.
2. Integrované vývojové prostředí, sdružující všechny prvky, které programátor při své práci nezbytně potřebuje.
3. Vizuální návrhář aplikací pro rychlý návrh grafického uživatelského rozhraní aplikace.
4. Profesionální editor pro zápis zdrojového kódu, který dokáže barevně odlišovat jednotlivé programové prvky a konstrukce, poradí si s automatickým doplňováním kódu a ve vhodné chvíli nabídne vývojáři seznam použitelných programových entit pomocí vestavěné inteligentní technologie.
5. Výkonný editor pro odladování (debugger), který dovede snadno a rychle odhalit chyby v programech, a to pokud možno již při psaní programového kódu. Editor by měl také umět krokovat programový kód a pracovat s programovými zarážkami.

Grafickou podobu vztahu mezi programovacími jazyky různých stupňů abstrakce můžete vidět na obrázku 1.1.