

Microsoft
.net Framework

**Příručka vývojáře
pro přechod na platformu Microsoft .NET**

Microsoft
Visual Studio.net

<http://msdn.microsoft.com/net/upgrading>

Přecházíme na platformu Microsoft .NET

Microsoft Corporation

POZNÁMKA PRO NAŠI VÝVOJÁŘSKOU KOMUNITU

S vydáním softwarového produktu Microsoft® Visual Studio® .NET 2003 jsme přivedli na svět již druhou verzi mocných a vzrušujících nástrojů pro budování a rozmísťování distribuovaných aplikací pro operační systémy Windows, Web a mobilní zařízení. Nejnovější vydání tohoto softwarového produktu je postaveno na všech výtečných vlastnostech našich předcházejících produktů a zahrnuje nové a vskutku průkopnické schopnosti a atributy, přesně také, kterých jste se dožadovali. Abychom vám pomohli lépe a snáze nalézt cestu ke všem novým technologiím, které vývojová platforma Microsoft .NET zastřešuje, připravili jsme pro vás tuto příručku. Kromě toho vám ukážeme, jak provést migraci vašich dosavadních znalostí a aplikací tak, aby byl váš přechod na platformu Microsoft .NET co možná nejefektivnější.

Dokumenty a dva CD disky, které jsou součástí této příručky, obsahují informace, které budete potřebovat proto, abyste z produktu Visual Studio .NET a platformy .NET Framework získali co nejvíce:

- Příručku tvoří některé z našich nejoblíbenějších a vysoce hodnocených dokumentů ze série *Přecházíme na platformu Microsoft .NET*. Naleznete zde migrační průvodce pro mnohé programovací jazyky a technologie. Kromě jiných se dozvíte důležité informace o přechodu na technologii ASP.NET, či o programovacích jazycích jakými jsou Visual Basic .NET, řízené (managed) C++ a C#. Stranou ovšem nezůstanou ani programátoři pracující v jazyce Java, neboť pro ně je připravena samostatná sekce s názvem *Přenos Java aplikací na platformu .NET Framework*. V neposlední řadě se můžete těšit také na vyčerpávající informace o ADO.NET, COM Interop a společném běhovém prostředí (Common Language Runtime, CLR).
- První CD disk, označený jako *Upgrading to Microsoft .NET (Inovujeme na Microsoft .NET)*, obsahuje úplnou kolekci materiálů, které vás uvedou do problematiky platformy .NET Framework, technologie ASP.NET a produktu Visual Studio .NET. Najdete zde také studie o vytváření tříd, práci s frontami zpráv a používáním XML webových služeb namísto DCOM. Většina těchto článků se zaměřuje především na vývojáře používající jazyk Visual Basic, ovšem vykládanou problematiku lze uplatnit také v dalších programovacích jazycích Visual Studia .NET. Pokud si chcete prohlédnout seznam všech položek, které jsou na CD disku umístěny, nalistujte stranu 194.

Na prvním CD disku můžete dále najít sadu cvičení a postupů, které vysvětlují proces provádění inovací aplikací z Visual Basicu 6.0 do Visual Basicu .NET a z Javy/JSP do C#/ASP.NET. Tyto materiály pro vás připravila společnost ArtinSoft, která je kromě jiného známá také tím, že její pracovníci vytvořili nástroje **Visual Basic Upgrade Wizard** a **Java Language Conversion Assistant**. Cvičení představují skvělou pomůcku pro získání základních znalostí o uvedených nástrojích, procesu konverze programového kódu různých programovacích jazyků a charakteristice nejběžnějších migračních scénářů. Ukázkové postupy vám pomohou urychlit proces vzdělávání a také vám ukáží, jak může být proces portace kódu aplikací efektivnější a produktivnější. Jakmile budete vybaveni patřičnými znalostmi .NET platformy a postupy pro inovaci vašich aplikací, budete moci získané dovednosti aplikovat i v mnoha jiných aplikacích. Navíc, vaše aplikace budou s podporou .NET platformy robustnější než kdykoliv předtím.

- Na druhém CD disku se nachází nový *Resource Kit* pro jazyk Visual Basic .NET, jenž představuje hodnotnou a vývojáři oblíbenou sadu cenných zdrojů pro programátory ve Visual Basicu. Visual Basic .NET Resource Kit je naplněn skutečně obrovským množstvím kvalitních .NET komponent třetích stran, které jsou vám poskytovány zcela zdarma. S Resource Kitem obdržíte také kompletní řešení ukázkových aplikací určených pro Windows i Web. Zdrojový kód ukázkových aplikačních řešení můžete použít i v jiných aplikacích, můžete jej modifikovat podle svých potřeb, nebo jej můžete použít pro studijní účely. Kdybychom chtěli popisovat všechny elementy, které tvoří Visual Basic .NET Resource Kit, jistě bychom zaplnili ještě mnoho stránek, proto se k dalším součástí této sady vyjádříme jenom stručně: Čekají na vás stovky ukázek zdrojových kódů aplikací, tréninková videa, zajímavé články a studie, ukázkové kapitoly ze znamenitých odborných publikací a mnohem více.

Naším cílem je podpořit a povzbudit vývojáře, programátory a softwarové architekty, aby pomocí softwarového produktu Visual Studio .NET začali vytvářet novou generaci počítačových aplikací. Již nyní se nemůžeme dočkat všech těch nových a vzrušujících řešení, které dokážete vyvinout prostřednictvím platformy .NET Framework. Doufáme, že tato příručka je dobrým startovním bodem. Budete-li chtít získat podrobnější informace o procesu inovace na novou vývojovou platformu, navštivte prosím <http://msdn.microsoft.com/net/upgrading>.

S úctou

Marie K. Huwe
General Manager
Microsoft .NET Platform a Evangelism Group

| | |
|----------------------------------------------|---|
| Poznámka pro naši vývojářskou komunitu | 2 |
|----------------------------------------------|---|

Kapitola 1: PROGRAMOVACÍ JAZYKY SPOLEČNOSTI MICROSOFT

| | |
|----------------------------------------|----|
| Přínosy platformy Microsoft .NET | 8 |
| Visual Basic .NET | 9 |
| Visual J# .NET | 16 |
| Shrnutí | 17 |

Kapitola 2: PROČ ASP.NET?

| | |
|---------------|----|
| Shrnutí | 20 |
|---------------|----|

Kapitola 3: PŘECHOD Z ASP NA ASP.NET

| | |
|----------------------------------------------------|----|
| Shrnutí | 24 |
| Bližší pohled na ukázkovou ASP aplikaci | 25 |
| Zkoumání programového kódu ukázkové aplikace | 26 |
| Převádění ASP aplikace do prostředí ASP.NET | 28 |

Kapitola 4: SMĚREM K ASP.NET: KLÍČOVÉ FAKTORY

| | |
|-----------------------------------------------------------------------|----|
| Shrnutí | 36 |
| Úvod | 36 |
| Potíže s kompatibilitou | 37 |
| Změny uvnitř základního aplikačního programového rozhraní (API) | 37 |
| Strukturální změny | 38 |
| Změny v syntaktické skladbě programovacího jazyka Visual Basic | 40 |
| Změny související se zabezpečením aplikací | 46 |
| Změny při přístupu k datům | 47 |
| Příprava pro přechod směrem k ASP.NET | 47 |
| Shrnutí | 49 |
| O autorovi | 49 |

Kapitola 5: VYLEPŠOVÁNÍ DOKONALOSTI: BUDOVÁNÍ APLIKACÍ PRO SYSTÉM WINDOWS S JAZYKEM VISUAL BASIC .NET

| | |
|---------------------------------------------------------------|----|
| Shrnutí | 52 |
| Úvod | 52 |
| Aplikace Hypotekární kalkulačka | 53 |
| Začínáme | 53 |
| Rozvržení formuláře | 54 |
| Nabídky | 59 |
| Pořadí, v jakém získávají instance zaměření (Tab Order) | 62 |
| Modifikace rozměrů | 63 |
| Psaní programového kódu | 64 |
| Sestavení projektu | 66 |
| O krok dále | 66 |
| Závěr | 66 |

Kapitola 6: OBJEKTIVĚ ORIENTOVANÉ PROGRAMOVÁNÍ V JAZYCE VISUAL BASIC .NET

| | |
|---------------------------------------------------------------------|----|
| Shrnutí | 68 |
| Úvod | 68 |
| Objektově orientované programování | 68 |
| Objektově orientované programování v jazyce Visual Basic .NET | 69 |
| Definice třídy | 69 |
| Vytváření konstruktorů a destruktorů | 71 |
| Vytváření a likvidování objektu | 72 |
| Použití třídy System.Object | 73 |
| Porozumění dědičnosti | 74 |
| Závěr | 76 |

Kapitola 7: JAK NA LADĚNÍ APLIKACÍ V JAZYCE VISUAL BASIC .NET

| | |
|----------------------------------------------------------|----|
| Shrnutí | 78 |
| Úvod | 78 |
| Vytváření kalkulačky životního pojištění | 78 |
| Nabídka Debug | 81 |
| Robustní okno Command | 81 |
| Okno Command: Mód Immediate | 82 |
| Okno Output | 83 |
| Zarážky | 84 |
| Opravujeme aplikaci Kalkulačka životního pojištění | 85 |
| Kde jsem byl? | 87 |
| Použití tříd Debug a Trace | 88 |
| Závěr | 90 |

Kapitola 8: DATOVÉ VAZBY A OBJEKTY PRO PŘÍSTUP K DATŮM V KNIHOVNĚ WINDOWS FORMS

| | |
|--------------------------------------------------------------------|-----|
| Shrnutí | 92 |
| Jednoduchá ukázka interakce s daty v prostředí Windows Forms | 93 |
| Oznamování změn vlastností | 94 |
| Vytváření vazeb s typově silnou (strongly-typed) kolekcí | 97 |
| Implementace rozhraní IBindingList | 100 |
| Notifikace změny | 101 |
| Editování a odstraňování položek | 102 |
| Přidávání položek | 102 |
| Implementace rozhraní IEditableObject v odvozených třídách | 103 |
| Kopírování dat | 103 |
| Zjišťování, zdali jsme noví | 104 |
| Metoda BeginEdit | 104 |
| Metoda EndEdit | 104 |
| Metoda CancelEdit | 105 |
| Zpracovávání události RemoveMe | 105 |
| Implementace rozhraní IDataErrorInfo v odvozených třídách | 106 |
| Závěr | 107 |

Kapitola 9: C++ → C#: CO POTŘEBUJETE VĚDĚT PRO PŘECHOD Z C++ K C#

| | |
|-------------------------------------------------|-----|
| Shrnutí | 110 |
| Přecházíme do řízeného prostředí | 110 |
| Pasti, nástrahy a léčky | 111 |
| Odkazové a hodnotové datové typy | 111 |
| Struktury | 112 |
| Vše je odvozeno od základní třídy System.Object | 112 |
| Použití klíčového slova new | 114 |
| Vlastnosti | 115 |
| Pole | 117 |
| Rozhraní | 119 |
| Rozhraní IEnumerable | 120 |
| Použití základní knihovny tříd | 121 |
| Čtení dat souboru přes síť | 124 |
| Vytváření klienta | 124 |
| Atributy a metadata | 125 |
| Uživatelské atributy | 126 |
| Použití atributu | 127 |
| Reflexe | 129 |
| Odhalování typů | 130 |
| Objevování typů | 131 |
| Vyhledávání členů | 131 |
| Získávání pouze metod | 132 |
| Získávání specifických členů | 132 |
| Dynamické spouštění metod | 133 |
| Závěr | 135 |
| O autorovi | 135 |

Kapitola 10: NATIVNÍ A ŘÍZENÉ C++: JAZYKOVÁ INTEROPERABILITA

| | |
|---------------------------------------------|-----|
| Dědictví | 138 |
| Jaké jsou vaše volby a kompromisy? | 139 |
| Bezpečný přístup ke kódu a ověřitelný kód | 140 |
| COM Interop | 141 |
| Vytváření COM komponenty | 141 |
| Použití COM komponenty z neřízeného kódu | 142 |
| Použití COM komponenty z řízeného kódu | 142 |
| Je COM Interop pro vás tou správnou volbou? | 144 |
| P/Invoke | 144 |
| Vytváříme knihovnu DLL | 144 |
| Použití knihovny DLL – Dřívější přístup | 145 |
| Použití knihovny DLL – Nový přístup | 145 |
| Kdo potřebuje P/Invoke? | 146 |
| V čem je rozdíl? | 147 |
| Portace pomocí XCOPY | 147 |
| Řízený obal | 149 |
| O autorce | 150 |

KAPITOLA 11: QUAKE II .NET

| | |
|-------------------------------------|-----|
| Zdrojový kód a soubory | 153 |
| Jak spustit Quake II .NET | 153 |
| Jak sestavit kód | 154 |
| Jak jsme přenášeli kód | 154 |
| Portace kódu do nativního C++ | 155 |
| Portace kódu do řízeného C++ | 156 |
| Rozšiřování hry Quake II | 157 |
| Výkonnost | 161 |
| Závěr | 162 |

KAPITOLA 12: PŘENOS JAVA APLIKACÍ NA PLATFORMU .NET FRAMEWORK

| | |
|---------------------------------------------------|-----|
| Shrnutí | 164 |
| Co znamená .NET? | 164 |
| .NET Framework | 164 |
| Visual Studio .NET | 166 |
| Porovnání .NET a Javy | 166 |
| Výhody platformy .NET Framework oproti Javě | 166 |
| Jak přenést Java aplikace do .NET | 168 |
| Mapování technologie | 170 |
| Migrační mix | 172 |
| Závěr | 180 |

KAPITOLA 13: PŘENOS NATIVNÍHO KÓDU DO ŘÍZENÉHO PROSTŘEDÍ PLATFORMY .NET FRAMEWORK

| | |
|-----------------------|-----|
| P/Invoke | 182 |
| Konverze typů | 184 |
| RCW a CCW | 187 |
| TLBIMP a TLBEXP | 187 |
| REGASM | 189 |
| O autorovi | 191 |

DODATEK: OBSAH CD „PŘECHÁZÍME NA .NET“ **194**

| | |
|----------------------|-----|
| Autor překladu | 199 |
|----------------------|-----|

1

Programovací jazyky společnosti Microsoft

Prashant Sridharan
Senior Product Manager
Microsoft Corporation

PŘÍNOSY PLATFORMY MICROSOFT .NET

Vývojová platforma Microsoft® .NET Framework je integrální součástí podporovaných operačních systémů Microsoft Windows® pro vytváření a běh nové generace aplikací a XML webových služeb. Zajišťuje vysoce produktivní, na prověřených standardech založené prostředí, v němž může společně koexistovat více programovacích jazyků pro snadnější vývoj různorodých počítačových aplikací. .NET Framework umožňuje vývojářům využívat již nabyté znalosti, protože výrazně usnadňuje proces programové integrace s již existujícími softwarovými aplikacemi, moduly či komponenty. Programátoři budou zcela jistě nadšeni také nepřehlédnutelnými inovacemi při vývoji a udržování internetových aplikací či při vytváření distribučních aplikačních jednotek. Vývojová platforma .NET Framework je postavena na dvou základních pilířích, jimiž jsou: společné běhové prostředí (Common Language Runtime, CLR) a jednotná a hierarchicky uspořádaná knihovna tříd, která zapouzdřuje třídy pro vývoj webových aplikací pomocí ASP.NET, inteligentních klientských aplikací pro Windows (pomocí Windows Forms) a databázových aplikací prostřednictvím subsystému ADO.NET.

Programátoři mohou psát své aplikace pro platformu .NET Framework v mnoha programovacích jazycích. Ať již použijete kterýkoliv z podporovaných jazyků, finálním výstupem kompilátoru bude vždy jazyková mezivrstva, která je tvořena kódem jazyka Microsoft Intermediate Language (MSIL). Kód jazyka MSIL je spravován společným běhovým prostředím, které vydá v příhodné chvíli pokyn na jeho překlad do nativního kódu, jemuž rozumí instrukční sada procesoru počítače. Jelikož výsledním produktem kompilátoru každého programovacího jazyka platformy .NET Framework je kód jazyka MSIL, aplikace napsaná v kterémkoliv jazyce může bez potíží spolupracovat s aplikací, která byla vytvořena v jiném programovacím jazyce. Poté, co bylo uvedeno společné běhové prostředí (Common Language Runtime), je jenom na programátorech, který programovací jazyk si z bohaté palety dostupných jazyků vyberou. Jednoduše řečeno, programátoři si teď mohou vybrat „ten svůj“ jazyk, a to plně podle svých preferencí a charakteru řešených úkolů.

Bohatá paleta programovacích jazyků

Podobně jako jsou umělci jedineční v používání materiálů a nástrojů, které vyhovují jejich zkušenostem a osobním preferencím, tak i softwaroví vývojáři uplatňují programovací jazyky v závislosti od svých znalostí a řešené problematiky. Žádný jazyk dosud nevyhověl potřebám všech vývojářů a programátorů. Po pravdě řečeno, programátoři jsou již od narození podivná stvoření: Zčásti jsou to umělci, zčásti vědci, pokaždé tvrdohlaví a věčně nespokojní. Přestože neustále hledají cesty k dosažení dokonalosti, jsou nuceni akceptovat jistou formu nedostatečnosti moderních programovacích jazyků. Pro některé z nich není problémem změnit vývojový nástroj jenom z rozmaru, zatímco jiní se snaží získat větší zkušenosti v rámci jednoho programovacího jazyka. Koneckonců, také Michelangelo byl talentovaným sochařem a brilantním malířem, kdežto Monet exceloval „pouze“ s olejovými barvami a plátnem. I tak jsou to oba géniové, kterých si vážíme.

„Vždycky plánujeme, že budeme mít půl milionu nebo 50 milionů oddaných programátorů ve Visual Basicu. Ale víte co? Máme Visual Basic v .NET. A teď máme i Javu v .NET. Dokonce máme i COBOL v .NET!“

– Tim Huckaby, President a CEO, Interknowldy

Výběr programovacího jazyka je v podstatě osobní volbou, která je ovlivněna větším počtem faktorů. Místo toho, abychom programátory nutili k osvojení jednoho jazyka před jiným, Microsoft nabízí platformu, na níž mohou různé programovací jazyky jako C++, Fortran, COBOL, Visual Basic® a Perl, společně prosperovat a umožňovat přístup k síle a flexibilitě platformy .NET Framework.

Co se programovacích jazyků týče, Microsoft nabízí čtyři zástupce s odpovídajícími vývojovými prostředími. Každý jazyk byl navržen tak, aby mohl oslovit vybranou školu programátorů. Jedná se o následující programovací jazyky:

- **Visual Basic .NET** - nejnovější verze světově nejpopulárnějšího vývojového nástroje a jazyka. Visual Basic .NET doručuje nepřekonatelnou produktivitu a jedinečné jazykové vlastnosti pro úkolově-orientované programátory, kteří budují svá řešení pomocí platformy .NET Framework.
- **Visual C++® .NET** - programovací nástroj, pro který je typická maximální síla a ovladatelnost. Pomocí jazyka C++ mohou programátoři, orientující se zejména na sílu jazyka, překlenout technologie různých platform a sestavovat tak nativní aplikace pro Windows, jako i moderní řešení pro platformu .NET Framework ruku v ruce s maximálním výkonem a rozšířenou funkcí.
- **Visual C#® .NET** - moderní a průkopnický programovací jazyk a nástroj. C# byl představen v roce 2001 a již od první chvíle poskytoval známou jazykovou syntaxi pro programátory v C++ a Javě, společně s unikátními jazykovými konstrukcemi, jež nabídly vývojářům pohled na elegantnější stavbu aplikací pro platformu .NET Framework.
- **Visual J#® .NET** - programovací jazyk, jenž je zaměřen na programátory v Javě, kteří by rádi rozšířili své pole působnosti a vyvíjeli řešení pro platformu Microsoft .NET. Visual J# .NET zprostředkovává jazyku Java a současným vývojářům ve Visual J++ úplný přístup ke zdrojům platformy .NET Framework a pokročilým XML webovým službám, zatímco pořád zachovává známé jazykové konstrukce a syntaxi.

Tato kapitola se zaměřuje na bohatou paletu programovacích jazyků, které vytvořila společnost Microsoft a které jsou dostupné pro vývojáře a programátory. Společně s platformou Microsoft .NET mohou programátoři využívat již získané schopnosti a dovednosti, zatímco jim nic nebrání v ovládnutí nové vývojové platformy, nástrojů a programovacích jazyků. Cílem je pomoci všem vývojářům vyvíjet podmanivá řešení rychleji než kdykoliv předtím.

VISUAL BASIC .NET

<http://msdn.microsoft.com/vbasic>

Visual Basic 1.0 způsobil zásadní převrat ve vývoji aplikací pro operační systém Windows odstraněním vstupních bariér a demonstrací, která názorně předvedla, že široké masy programátorů mohou být ve své práci daleko produktivnější. Také nejnovější reinkarnace s názvem Visual Basic .NET staví na kladných vlastnostech všech svých předchůdců. Visual Basic .NET vsází na intuitivní syntaxi a uživatelské rozhraní, stejně jako na množství podpůrných nástrojů a migračních průvodců, kteří urychlují sestavování aplikací napojených na platformu Microsoft .NET. Visual Basic .NET s výhodou využívá mimořádnou efektivnost vývojového procesu, což je vlastnost, jež byla již mnohokrát obhájena také stylem práce předcházejících verzí tohoto programovacího nástroje. Samozřejmě, nová edice Visual Basicu přidává ještě spoustu nových a ještě hodnotnějších programových funkcí, které dávají programátorům všech úrovní, od začátečníků až po zkušené vývojáře, možnost budovat aplikace pro Windows, Web a mobilní zařízení.

Vývoj orientovaný na úkoly

Závazné termíny nejsou ve světě softwarového průmyslu ničím neznámým. Ve skutečnosti jsou závazné termíny pro mnoho programátorů na denním pořádku. Tito programátoři bývají často pověřováni vývojem oportunistických aplikací, které řeší určité obchodní požadavky. Aplikace přitom vyžadují jenom malé množství plánování a rychlou cestu k tvorbě distribučních jednotek. V některých případech bývají takováto řešení pečlivě testována a odlaďována, zatímco jindy je primárním cílem pouhé sestavení aplikace a vytvoření příslušných aplikačních souborů určených k distribuci, čímž je úkol programátora splněn a ten se pak může vrhnout na další z nastávajících úkolů. Tato skupina vývojářů se označuje jako vývojáři orientovaní na úkoly, jejichž cílem je rychlé doručení vyvinutého řešení. Z tohoto důvodu je nutné, aby úkolově orientované vývojové nástroje kladly důraz na vysokou produktivitu práce, a to prostřednictvím vytvoření abstraktní vrstvy mezi programátorem a níže umístěnou vývojovou platformou. Programátor se poté může soustředit především na návrh aplikace, její vývoj a flexibilní rozmístění aplikace mezi cílovými skupinami uživatelů.

Programátorova volba

Visual Basic .NET představuje ideální variantu pro následující typy programátorů, kteří chtějí používat vývojovou platformu .NET Framework pro vytváření nové generace aplikací a softwarových služeb:

- **Programátoři hledající rychlý a produktivní vývojový nástroj pro platformu .NET Framework.** Microsoft Visual Basic .NET nabízí snadno pochopitelnou jazykovou syntaxi a intuitivní vývojové prostředí, které v notné míře pomáhá programátorům při návrhu, programování a sestavování jejich aplikací. Navíc, kolem Visual Basicu se vytvořila již velice početná vývojářská komunita s takřka nevyčerpatelnými zdroji informací, které jsou vývojářům ve Visual Basicu .NET plně k dispozici. Pomocí těchto informačních zdrojů mohou vývojáři citelně urychlit svůj start do tajů nové platformy Microsoft .NET.
- **Programátoři, kteří jsou znalí dřívějších verzí jazyka Visual Basic.** Visual Basic .NET staví na klíčových slovech, syntaxi a dalších známých elementech jazyka Visual Basic. Kupříkladu implicitní nerozlišování malých a velkých abecedních písmen či snadno pochopitelná jazyková syntaxe bude pro tradiční programátory ve Visual Basicu dobře známá již od prvních okamžiků v novém prostředí. Pokud jste investovali mnoho úsilí do vývoje řešení v předcházejících verzích jazyka Visual Basic, můžete přenést kód těchto projektů i do nové verze, a to pomocí vestavěného průvodce pro inovaci. Nicméně, většina existujících ActiveX[®] ovládacích prvků může být i ve Visual Basicu .NET nadále používána.
- **Vývojáři požadující známá paradigma, která se týkají vizuálního vývoje aplikací a inteligentní práce s editorem pro zápis programového kódu.** Mnohé principy vytváření vizuálního rozhraní aplikací v režimu jejich návrhu a psaní zdrojového kódu byly převzaty z předchůdců Visual Basicu .NET a dále rozšířeny. To znamená, že i v novém Visual Basicu se střetnete s vizuálním návrhem grafického rozhraní aplikací a s technologií IntelliSense[®], která vám bude asistovat při psaní kódu. Pochopitelně, vývojáři jazyka nezapomněli ani na implementaci automatického formátování programového kódu, což je vskutku užitečná vlastnost, která zabezpečuje mnohem snazší čitelnost napsaného zdrojového kódu aplikací.
- **Programátoři, kteří by rádi budovali své aplikace pomocí přívětivějšího a přístupnějšího programovacího jazyka.** Visual Basic .NET byl zkonstruován tak, aby byl otevřený široké škále vývojářů, od nováčků až po experty. Začátečníci zanedlouho objeví mnoho vynikajících programových vlastností a charakteristik, které jsou stejně užitečné jako klíčová vylepšení prostředí Visual Basicu.

Jedinečné jazykové dovednosti

Programovací jazyk Visual Basic .NET obsahuje několik zcela výjimečných jazykových rysů a vlastností, které nelze najít v jiných programovacích jazycích společnosti Microsoft pro vývojovou platformu .NET. Zde je jejich výčet:

- **Implicitní inicializace proměnných.** Visual Basic .NET nepožaduje, aby byly proměnné a datové členy inicializovány před svým použitím. Začínající programátoři proto nebudou zbytečně uváděni do rozpaků zdánlivě tajemně vyhlížejícími požadavky, které můžete najít v jiných .NET programovacích jazycích.
- **Implicitní přiřazování datových typů proměnným a pozdní vázání objektů (late-binding).** Ve zdrojovém kódu jazyka Visual Basic .NET nemusí být explicitně specifikován datový typ proměnné před jejím použitím, což pomáhá programátorům psát užitečný kód již po krátkém tréninku.
- **Způsob práce enumerací.** Visual Basic .NET nabízí programátorům intuitivnější chování kódu při práci s enumeračními (výčtovými) typy.
- **Implicitně veřejný přístup.** Implicitně jsou vlastnosti a metody tříd napsaných v jazyce Visual Basic .NET veřejně přístupné. Tato skutečnost přispívá k přátelštější a intuitivnější programové syntaxi.
- **Použití sdílených členů.** Přístup ke sdíleným členům lze ve Visual Basicu .NET provést buď pomocí názvu třídy, nebo názvu instanční proměnné příslušného typu, jemuž sdílené členy patří. Zápis programového kódu je tím pádem snazší a názornější. Kupříkladu:

```
Dim x As New MyClass
x.SharedMethod() ' Pracuje stejně dobře jako kód na dalším řádku.
MyClass.SharedMethod()
```

- **Volitelné parametry.** Visual Basic .NET podporuje volitelné parametry, pomocí nichž mohou návrháři tříd pracovat flexibilněji při sestavování svých knihoven tříd. Volitelné parametry rovněž poskytují programátorům schopnost psát prospěšný kód, aniž by bylo nutné učit se všechny prvky objektově orientovaného programování.
- **Filtrované Catch bloky.** Visual Basic .NET zavádí novou a nutno přiznat, že značně efektivní strukturovanou správu chybových výjimek prostřednictvím filtrovaných **Catch** bloků. Filtrované **Catch** bloky umožňují vývojářům zachycovat a filtrovat chyby na základě třídy výjimky, jakéhokoliv podmíněného výrazu nebo explicitního identifikačního čísla chyby.
- **Parametrické vlastnosti.** Vlastnosti ve Visual Basicu .NET mohou obsahovat parametry, a tudíž jsou flexibilnější než jejich protějšky z C#.
- **Deklarativní zpracovatele událostí.** Zpracovatele událostí ve Visual Basicu .NET mohou deklarovat události, na které mohou reagovat pomocí klíčového slova **Handles**.
- **Redeklarace členu rozhraní.** Visual Basic .NET dovoluje programátorům přejmenovat člen rozhraní při jeho implementaci ve třídě.

Unikátní dovednosti prostředí Visual Basicu .NET

Visual Basic .NET kromě početných jazykových rozšíření přichází také s klíčovými inovacemi svého prostředí, které nyní vychází programátorům a vývojářům ještě více vstříc a umožňuje jim navrhovat a psát okouzující aplikace a softwarové služby. Tyto inovační prvky jsou dostupné výhradně ve Visual Basicu .NET:

- **Kompilace v pozadí.** Kompilace v pozadí pracuje permanentně za scénou a tiše kompiluje všechny programový kód, který zadáte do editoru. Programátoři ve Visual Basicu .NET tak mohou okamžitě vědět, zdali je zapsaný kód bez chyb, nebo se v něm nacházejí nějaké nesrovnalosti.
- **Pokročilé formátování kódu.** Editor pro zápis programového kódu může (volitelně) automaticky formátovat zapsaný kód, čímž výrazně šetří váš čas. Pokročilé formátování si dovede poradit s automatickým zarovnáváním fragmentů zdrojového kódu, se změnou velkých či malých abecedních znaků v názvech klíčových slov a proměnných či dokonce s přidáním chybějícího návěstí `Then` v příkazu `If` a mnoha dalšími operacemi.

Výkon

Finální oblastí s velkou mírou důležitosti je výkon. Kompilátor Visual Basicu .NET generuje výsledný kód v podobě kódu jazyka Intermediate Language (IL), který je, co se týče výkonu, plně srovnatelný s kódem, jenž je generován kompilátorem jazyka C#.

VISUAL C++ .NET

<http://msdn.microsoft.com/visualc>

Programátoři, pro které je podstatný především výkon jejich aplikací, budou zcela určitě potřebovat vytěžit z vývojové platformy všechny její schopnosti, a to při maximálním možném výkonu. Navzdory tomu, že platforma .NET Framework a společné běhové prostředí přináší mnoho výhod, někteří programátoři budou i nadále vytvářet aplikace, jejichž činnost bude využívat veškeré pokročilé programové funkce operačního systému Windows. Tento segment vývojářů pracuje již tradičně s jazykem Visual C++, pomocí něhož mohou dosáhnout preciznější kontroly nad během aplikací. Programátoři mohou aplikovat různé metody, jako je například hloubková optimalizace programového kódu a velice efektivní přístup k systémovým službám, jako je třeba přístup do operační paměti či do souborové struktury pevného disku. Visual C++ .NET v tomto směru následuje svého předchůdce a také nabízí softwarovým odborníkům přímý přístup k aplikačnímu programovému rozhraní Win32® API. Visual C++ ovšem obsahuje také přidanou hodnotu v podobě zcela neomezeného přístupu k bohaté nabídce dovedností vývojové platformy .NET Framework a řízení aplikací prostřednictvím společného běhového prostředí.

Vývoj orientovaný na výkon

Existuje opravdu nepřehledný počet situací, kdy musí programátoři sáhnout po vybraných fundamentálních programových funkcích operačního systému. Z historického hlediska uváděla společnost Microsoft na trh spektrum programovacích nástrojů, z nichž některé prováděly abstrakci od funkcí operačního systému v plném rozsahu, zatímco jiné doručovaly ničím nespoutaný přístup ke všem programovým zákoutím operačního systému. V současné době je situace taková, že vývojová platforma .NET Framework nabízí řadu aplikačních programových rozhraní pro programování robustních podnikových řešení, ovšem přesto nezajišťuje přístup k veškerým funkcím, které můžete najít v operačních systémech řady Microsoft Windows. Výkonově orientované vývojářské nástroje berou v potaz požadavek vývojářů na maximální výkon aplikací, a proto přicházejí s pokročilými jazykovými konstrukcemi, knihovnamí tříd a prostředími navrženými tak, aby byl zaručen komfortní vývoj plně škály řešení, která odpovídají požadavkům zákazníků.

Programátorova volba

Visual C++ .NET je vhodným vývojovým nástrojem pro mnoho programátorů:

- **Programátoři, kteří si přejí budovat aplikace a komponenty založené na Win32 API.** I v dnešních dobách moderních metod a technik vývoje softwaru žijí programátoři, kteří stále potřebují vyvíjet nativní aplikace pro systém Windows. Tato skupina programátorů používá při své práci aplikační programové rozhraní Win32 API a nativní knihovny tříd jazyka C++ pro dosažení mimořádného výkonu a funkčnosti. Visual C++ .NET 2003 uvádí některé optimalizace vestavěného kompilátoru, které mohou podpořit programátory při dosahování ještě větších výkonnostních efektů v kýžených typech aplikací.
- **Programátoři, kteří by rádi překlenuli mezeru mezi nativními aplikacemi a aplikacemi vyhovujícími standardům platformy .NET Framework.** Existující aplikace mají často složitou programovou strukturu, vytvoření které stálo mnoho času, energie a finančních prostředků. Z mnoha důvodů není možné, aby byly tyto aplikace okamžitě přepsány pro běh na platformě .NET Framework. Pomocí nástroje Visual C++ mohou vývojáři rozšířit existující aplikace nebo přidat nový programový kód, jehož exekuci bude mít na starosti .NET Framework. Přitom však stále mohou přistupovat i k těm nejpokročilejším funkcím Windows API. Aktivovat programové funkce Win32 API mohou programátoři také z jiných jazyků jako je C# či Visual Basic, ovšem ani jeden z uvedených konkurentů nedisponuje vrozenou schopností C++ pro interoperabilitu s existujícím kódem systému Windows.
- **Vývojáři zaměřující se v první řadě na výkonnostní charakteristiky svých aplikací.** Jazyk C++ poskytuje vývojářům nejvyšší stupeň kontroly při návrhu a běhu programů. Zkušení programátoři mohou použitím jazyka C++ navrhovat a implementovat aplikace, jejichž kód je prováděn rychleji a efektivněji ve srovnání s tím, kdyby byly tyto aplikace napsány v jiných programovacích jazycích. To se týká jednak nativních aplikací určených pro operační systém Windows, jako i řízených .NET aplikací.
- **Programátoři, kteří chtějí vyvíjet skutečná meziplatformní řešení.** Jedině programovací jazyk C++ zahrnuje ISO standardizovanou a skutečně přenositelnou programovou syntaxi, kterou lze portovat na prakticky každý operační systém. Visual C++ .NET 2003 se daleko více přibližuje k stanoveným syntaktickým standardům, než tomu bylo kdykoliv předtím. Programátoři mohou těžit z pokročilých jazykových prvků a výhod, které nabízejí knihovny tříd pro širokou paletu rozmanitých operačních systémů.

Jedinečné jazykové dovednosti

Visual C++ .NET zahrnuje množství jedinečných jazykových rysů a dovedností, které byly do jazyka zařazeny pro uspokojení vysokých nároků náročných programátorů a softwarových expertů. Tyto jazykové schopnosti přispívají ke skutečnosti, že C++ je nejmocnější ze všech programovacích jazyků softwarového produktu Visual Studio společnosti Microsoft. Na následujících řádcích je uvedena charakteristika elementů, které dělají jazyk C++ tak výjimečným:

- **Šablony.** Šablony jsou charakteristické několika jazykovými rysy v době kompilace programu, které jsou dostupné jenom v C++. Používání šablon představuje dobrý krok pro zabezpečení znovupoužitelnosti a lepší výkonnosti programového kódu.
- **Ukazatele.** Pomocí ukazatelů mohou programátoři získat přímý přístup do různých částí operační paměti počítače. Ukazatele jsou nepostradatelnou programovací pomůckou, prostřednictvím které lze maximalizovat rychlost exekuce programových instrukcí aplikací.
- **Vícenásobná dědičnost.** Jazyk C++ uvádí programátory do vzrušujícího světa objektivě orientovaného programování (OOP) tím, že implementuje všechny důmyslné prvky a konstrukce této filozofie programování.
- **Rozšiřující sada vnitřních API (intrinsics).** Vývojáři mohou nyní pracovat s klíčovými prvky platformy, které nejsou dostupné přes standardní programovací techniky. Příkladem je aktivace vnitřních instrukcí sady MMX nebo AMD 3DNow!.
- **Atributy v době kompilace programu.** Atributy v C++ reprezentují snadno dosažitelné prostředky pro vytváření vysoce odladěných opakujících se segmentů programového kódu využitím jednodušší a robustnější jazykové syntaxe.

Unikátní vlastnosti vývojového prostředí

Visual C++ .NET 2003 je obdařený spoustou inteligentních rysů vývojového prostředí, které asistují programátorovi při vytváření flexibilních a výkonných aplikací:

- **Optimalizovaný kompilátor.** Kompilátor ve Visual C++ dovede vyladit aplikace podle různých scénářů, mezi nimiž nechybí optimalizace pro cílová běhová prostředí, optimalizace provádění kalkulací s čísly s pohyblivou řádovou čárkou či optimalizace generování kódu jazyka MSIL.
- **Kontrola bezpečnosti kódu za běhu programu.** Programátoři mohou psát bezpečnější nativní aplikace pro operační systémy Windows za pomoci pokročilých vlastností kompilátoru, které pomáhají ochraňovat aplikace před škodlivými útoky.
- **Podpora 32 a 64 bitů.** Kompilátory pro Visual C++ jsou dostupné pro velké množství hardwarových platform, mezi nimiž nechybí 32 a 64bitové mikroprocesory společností Intel a AMD jako i další mikroprocesorová zařízení. Psaní skutečně škálovatelných aplikací je tedy docela snadné.
- **Pokročilé hlášení programových chyb.** Není žádnou novinkou, že počítačové aplikace jsou náchylné na výskyt programových chyb. Visual C++ dovoluje vývojářům snadnější identifikaci a korekci chyb, a to dokonce i v distribučních jednotkách aplikací pomocí technologie Minidump.
- **Vyspělé techniky odladování aplikací.** Odladovací nástroje Visual Studia a Visual C++ jsou garancí současného odladování nativního i řízeného zdrojového kódu.

V budoucích verzích programovacího nástroje Visual C++ .NET se vývojáři mohou těšit na další nepostradatelná vylepšení a nové inovativní techniky, které posunou pomyslnou laťku pokroku ještě o kousek výše:

- **Generics.** Nová technologie, která nabídne opětovné použití parametrizovaných algoritmů za běhu programu.
- **Šablony řízených typů.** Nové šablony s sebou přinesou schopnost používat šablonovou syntaxi jazyka C++ s řízenými typy.
- **Bohatší podpora pro vytváření assembly, která budou těsněji provázána s pravidly společné jazykové specifikace (Common Language Specification, CLS).** Kompilátor jazyka C++ bude umožňovat programátorům označovat CLS-nekompatibilní typy a neověřitelný programový kód.

VISUAL C# .NET

<http://msdn.microsoft.com/vcsharp>

Programovací jazyky Visual Basic a Visual C++ stály odjakživa na rozdílných stranách spektra vývojářů. Visual Basic kladl důraz především na vysokou produktivitu práce a nabízel programátorům co možná nejlépe a nejpřátelštější vývoj aplikací, a to i za cenu poněkud omezeného přístupu k systémovým zdrojům. Na druhé straně, Visual C++ šel takřka na hranice možností, protože dovoloval programátorům převzít kontrolu nad systémem plně do svých rukou, i když někdy bylo nutné obětovat notnou dávkou pracovní produktivity.

Společnost Microsoft se rozhodla jednou a provždy překlenout pomyslný most mezi oběma uvedenými vývojovými prostředky, a proto navrhla a vyvinula moderní, flexibilní a inovativní programovací jazyk s názvem C#. C# je vhodným nástrojem pro vývojáře, kteří se soustřeďují zejména na eleganci a efektivnost napsaného zdrojového kódu. Ve skutečnosti budou konstrukce a syntaxe jazyka C# okamžitě známé všem vývojářům v C++. Přestože je C# dalším vývojovým stupínkem v evoluci jazyků C/C++, produktivitou své práce se může rovnat i s Visual Basicem.

Vývoj orientovaný na programový kód

Zcela jistě se shodneme na tom, že všichni programátoři musí při plnění svých projektů psát mnoho řádek programového kódu. Psaní kódu ovšem není jedinou činností, kterou je programátor živ. Kromě zadávání příkazů a klíčových slov tráví programátoři spoustu času také používáním průvodců, ovládacích prvků a návrhářů s cílem zvýšit produktivitu a efektivnost své práce v rámci návrhu a vývoje softwarových aplikací. Negativním efektem programátorů specializujících se na kód je skutečnost, že mohou postupně ztrácet kontrolu nad vším tím kódem, jenž vyplňuje jejich programy. Tito programátoři dokonce tak lpí na kódu, jenž vytvoří, že po jistém čase začnou věřit „svému“ kódu více než kódu, který byl připraven průvodci vývojového prostředí. A i když v některých okamžicích přece jenom sáhnou po průvodci či návrhář, pořád se budou snažit modifikovat vygenerovaný kód podle svých představ.

Kromě toho, programátoři zaměřující se na kód mají tendenci k psaní kódu, jenž bude opětovně použitelný jinými vývojáři, dokonce i takovými, kteří nemají úplné znalosti jejich vývojových praktik. Vývojová řešení a knihovny tříd, které jsou výsledkem práce těchto programátorů, vznikají především v editorech zdrojových kódů a ve většině případů postrádají implementaci pečlivější navrhovací fáze.

Programátorova volba

Pokud hledáte programovací jazyk vývojové platformy .NET Framework pro vytváření nové generace aplikací a služeb, C# je pro vás tou správnou volbou. Po pravdě řečeno, jazyk C# je vhodný pro všechny následující segmenty softwarových odborníků:

- **Programátoři hledající produktivní programovací jazyk z rodiny jazyků C/C++.** C# přichází s programovou syntaxí, která bude již na první pohled známá všem vývojářům, kteří jsou znalí jazyků C a C++. C# má však ve své nabídce daleko více skvělých prvků, mezi které patří podpora přetěžování operátorů, práce s enumeračními typy, rozlišování malých a velkých abecedních znaků, práce s vlastnostmi, událostmi, atributy, delegáty a další. Ruku v ruce s platformou .NET Framework jsou známé, ale i nově zavedené prvky jazyka rychle pochopitelné pro programátory v C++. Ti tak mohou okamžitě okusit vyšší efektivnost práce, lepší ovladatelnost a v neposlední řadě také signifikantní přínosy v oblasti bezpečné exekuce zdrojového kódu.
- **Návrháři systémů a softwaroví architekti.** Portfolio programových schopností jazyka C# je doslova přeplněno úžasnými rysy, mezi něž bezpochyby patří přetěžování operátorů či dokonce práce s nezabezpečeným kódem, pomocí něhož je možné komunikovat s aplikacemi staršího data. Všechny tyto výtečné vlastnosti dělají z jazyka C# kandidáta pro vývoj komplexních podnikových systémů a knihoven tříd. Inovace lze však očekávat i do budoucna: Nové prvky jako Generics a iterátory budou i nadále přispívat k tomu, abyste mohli pomocí jazyka C# vyvíjet mohutná podniková řešení na platformě .NET Framework.
- **Programátoři, kteří již investovali mnoho prostředků do vývoje softwarových produktů pomocí jazyka Java.** Patříte k programátorům s mnoha aplikacemi naprogramovanými v Javě, kterým jste věnovali mnoha času, energie a finančních prostředků? Chtěli byste svá stávající řešení portovat na platformu .NET Framework? Jestliže ano, programovací nástroj Visual C# .NET 2003 vám může podat pomocnou ruku. Součástí nástroje je totiž průvodce Java Language Conversion Assistant (JLCA), jenž je schopný provést rychlou konverzi předemných fragmentů programového kódu z Javy do C#. JLCA provádí pečlivou analýzu zdrojového kódu jazyka Java, který vzápětí převádí do kódu jazyka C#. Pokud dojde při konverzním procesu k potížím, jsou problémové partie kódu zřetelně označeny, takže programátoři je mohou rychle najít a dokončit konverzní proces.

C#: Programovací jazyk s mnoha prvky jazyka C++

Je to skutečně tak. Programovací jazyk C# přináší mnoho skvělých vlastností z dnes již tradičního jazyka C++ i do prostředí moderní vývojové platformy .NET Framework. Ano, spousta programových elementů a konstrukcí se podobá těm, které znáte z C++. V prostředí jazyka C# však můžete k těmto prvkům přistupovat s daleko větší efektivností a produktivitou práce. Na následujících řádcích se dozvíte více:

- **Podpora všech typů společného běhového prostředí.** Programovací jazyk C# dokáže pracovat se všemi typy, které může společné běhové prostředí použít.
- **Předávání argumentů odkazem a výstupní parametry.** Jestliže se rozhodnete pro C#, budete moci předávat proměnné funkcím odkazem anebo dokonce vytvářet výstupní parametry, které je nutno inicializovat před ukončením práce funkce, v níž jsou definovány.
- **Přetěžování operátorů.** Vývojáři řešení mohou vytvářet robustnější knihovny tříd pomocí techniky zvané přetěžování operátorů.
- **Příkaz using.** Aplikací příkazu using mohou programátoři lépe kontrolovat jak jejich aplikace hospodaří se zdroji.
- **Nezabezpečený kód.** Jazyk C# dovoluje vývojářům pracovat s ukazateli a manipulovat tak s vybranými segmenty operační paměti. Ačkoliv nezabezpečený kód je pořád realizován v rámci společného běhového prostředí, jeho použití umožňuje programátorům preciznější ovladatelnost využití systémové paměti. Navzdory tomu, pokud si přejete získat maximální možnou kontrolu nad operační pamětí, lépe vám bude pravděpodobně vyhovovat Visual C++.
- **XML dokumentace.** Programátoři v C# mohou opatřit vybrané partie zdrojového kódu poznámkami jazyka XML.

Je zcela zřejmé, že programovací jazyk C# je vskutku životaschopnou alternativou ve světě programování profesionálních aplikací. Dokonce jde o tak rychle se rozvíjející technologii, že již začíná přesahovat meze svého původního návrhu. Vývojáři a designéři jazyka C# plánují do budoucích verzí zařadit množství dalších zajímavých prvků a programových konstrukcí, které budou dále rozvíjet moderní a inovativní prostředek pro vývoj všemožných typů aplikací. A co se očekává?

- **Generics.** Obdoba šablon z C++, které výrazně přispějí k lepšímu opětovnému použití již jednou napsaného zdrojového kódu.
- **Iterátory.** Programová konstrukce, prostřednictvím které budete moci rychleji a snadněji pracovat s datovými kolekcemi.
- **Anonymní metody.** Zapojíte-li do kódu anonymní metody, můžete vykonávat jednoduché úkoly pomocí delegátů.
- **Parciální typy.** Implementace parciálních typů nabídne programátorům možnost rozdělit programový kód do více souborů.

VISUAL J# .NET

<http://msdn.microsoft.com/vjsharp>

Společnost Microsoft přivedla na svět také programovací jazyk Visual J# .NET, jehož cílem je nabídnout syntaxi známého programovací jazyka Java i pro vývojovou platformu .NET Framework. Visual J# .NET přijde vhod programátorům, studentům a profesorům, kteří mohou budovat svá řešení pro platformu .NET Framework, aniž by byli nuceni vzdávat se již zažitého programovacího stylu, jenž znají z Javy. Visual J# .NET ovšem oslovuje také mnohočetnou skupinu programátorů, kterým přirostl k srdci programovací nástroj Visual J++ 6.0. I tito vývojáři mohou využít svých dosavadních znalostí a začít s tvorbou nové generace aplikací a softwarových služeb pro moderní vývojovou platformu.

Vývoj v jazyce Java

Programovací jazyk Java vyřešil více problémů, kterým byli vývojáři v C++ nuceni často čelit, a to při zachování jednodušší syntaxe a známých prvků objektové orientovaného programování. Pro všechny vývojáře, kteří jsou znalci jazyka Java, je přechod na Visual J# .NET naprosto plynulý a po krátkém seznámení se s prostředím se zde budou cítit jako doma. Výběr jazyka Visual J# .NET navíc znamená, že programátoři mohou využívat všechny vzrušující vlastnosti, které jim .NET Framework nabízí.

Programátorova volba

Programovací nástroj Visual J# .NET je nepostradatelný pro následující typy počítačových specialistů:

- **Stávající programátoři pracující v jazyce Java.** Je zcela samozřejmé, že programátoři, kteří jsou velmi dobře obeznámeni s jazykem Java, se nebudou chtít pouštět do studia nového programovacího jazyka. Visual J# .NET staví na dosavadních zkušenostech vývojářů v Javě a výrazně tak minimalizuje investice nutné pro přechod na platformu .NET Framework.
- **Programátoři, kteří vynalozili prostředky na vývoj aplikací ve Visual J++.** Visual J# .NET dokáže konvertovat projekty vytvořené v prostředí Visual J++. Takto upravené projekty je okamžitě možné dále rozšiřovat a obohacovat o novou funkcionalitu dostupnou na platformě .NET Framework.
- **Studenti, učitelé a profesori.** Programovací nástroj Visual J# .NET mohou využívat studenti, učitelé a profesori pro výučbu základů počítačových věd. Visual J# .NET vyhovuje požadavkům zkoušky Advanced Placement Computer Science.

Jedinečné jazykové vlastnosti

Pro vývojáře pracující s jazykem Java bude přechod na J# velice snadný díky zařazení mnoha známých prvků a programových konstrukcí, které napomáhají tomu, aby se práce v jazyce J# stala synonymem pro komfortní a příjemný vývoj aplikací běžících pod křídly platformy .NET Framework. J# přichází s několika železky v ohni:

- **Syntaxe jazyka Java.** Pro programátory v Javě je nejdůležitějším zjištěním, že v prostředí J# naleznou jazykovou syntaxi, která je jim dobře známá. Dalším pozitivem je neomezený přístup ke zdrojům platformy .NET Framework.
- **Podpora knihoven tříd.** V jazyce J# mohou vývojáři pracovat s nezávisle vyvinutými knihovnami tříd, které poskytují takřka veškerou funkcionalitu knihoven tříd z Java Development Kit (JDK) verze 1.1.4 a také mnoho tříd z balíčků JDK 1.2 java.util.
- **Vlastnosti, delegáti a události.** Do programovacího jazyka J# byla začleněna podpora stěžejných prvků platformy .NET Framework jako jsou vlastnosti, delegáti či události. Kromě toho byla zachována soudržnost s tradiční syntaxí jazyka Java.
- **Javadoc komentáře.** J# si dokáže poradit s komentáři fragmentů zdrojového kódu podle stylu Javadoc. Visual J# .NET obsahuje nástroj, jenž umožňuje uživatelům vyvolat HTML API, které dokáže generovat dokumentaci z určených Javadoc komentářů.

Unikátní vlastnosti vývojového prostředí

Programovací jazyk Visual J# .NET je plně integrován do vývojového prostředí Visual Studio .NET, což umožňuje vývojářům v J# využívat vestavěné návrháře, odlaďovací nástroje či editory s pokročilými vlastnostmi. Rovněž byly vytvořeny také nové pomůcky, jejichž cílem je podat vývojářům pomocnou ruku při portaci jejich řešení na platformu Microsoft .NET:

- **Průvodce inovací pro Visual J++.** Programátoři používající Visual J++ mohou inovovat své projekty směrem k Visual J# .NET. Průvodce inovací provádí konverzi projektových souborů a nabízí uživatelům kontextuální nápovědu v případě potíží.
- **Binární konvertor.** Tento nástroj realizuje konverzi souborů s bajtovým kódem Javy do podoby assembly, kterou lze použít v .NET aplikacích.

SHRNUTÍ

Programovací jazyky mohou být využívány pro budování široké škály programových řešení. Každý jazyk je vybaven svými jedinečnými vlastnostmi, které z něj dělají výtečně navržený celek pro přípravu rozmanitých typů aplikací. Společnost Microsoft nabízí programátorům, vývojářům a softwarovým architektům paletu čtyř programovacích jazyků, které společně představují přístup k vývojové platformě .NET Framework pro mnoho milionů uživatelů.

2

Proč ASP.NET?

Microsoft Corporation
