

Zamřížovaná káva v síti

Microsoft Visual J# .NET verze 2003 je součástí Microsoft Visual Studia .NET verze 2003 (psali jsme o něm v Chipu 9/03). Protože se s předchozí verzí Visual Studia tento nástroj nedodával, podíváme se na něj podrobněji.

Tento článek vychází ze zkušeností s Visual J# z Visual Studia .NET 2003 Enterprise Architect.

O co jde a o co nejde

Nástroj Visual J# slouží pro vývoj aplikací pro platformu .NET a je určen vývojářům, kteří znají programovací jazyk Java, nejlépe Visual J++ 6.0. Obsahuje integrované vývojové prostředí (IDE) podobné jako ostatní programovací jazyky Visual Studia, překladač jazyka J#, překladač bajtového kódu Javy do mezijazyka MSIL a sadu nezávisle vyvinutých knihoven tříd, které poskytují možnosti odpovídající třídám z knihoven Javy. Podporuje také doplňkové balíky com.ms.lang, com.ms.dll, com.ms.com, com.ms.win32, com.ms.util a com.ms.jdbc.odbc z Visual J++, z nichž některé poskytují funkcionalitu obdobnou jako JDK 1.2. Visual J# není určen k vývoji aplikací pro JVM, i když jazyk J# je syntakticky shodný s Javou.

IDE

Už jsme si řekli, že IDE Visual J# je shodné s IDE ostatních programovacích jazyků Visual Studia. To znamená, že obsahuje editor, který umožňuje číslovat řádky, zvýrazňuje syntax, poskytuje nástroje pro ladění, bublinovou nápovědu při psaní, umožňuje vizuální vývoj aplikací s grafickým uživatelským rozhraním atd. Vedle toho poskytuje řadu oken, která nabízejí různé pohledy na aktuální projekt, dynamickou nápovědu apod. O tom všem jsme podrobněji hovořili v článku o Visual Studiu .NET 2003. Poznamenejme však, že dynamická nápověda k Visual J# nabízí pouze informace o některých knihovních třídách, nikoli o jazyku J# nebo o .NET Framework SDK. Informace je nutno čerpat především z webových stránek firmy Microsoft.

Jazyk J#

Už víme, že J# je programovací jazyk prakticky shodný s Javou. Přesněji, po stránce syntaxe jde o Javu na úrovni JDK 1.1.4 s microsoftskými rozšířeními, jako jsou delegáty a J/Direct. (Připomeňme, že delegáty jsou objektově zapouzdřené ukazatele na metody objektových typů, které se svého času staly příčinou nejprve slovních a pak i soudních sporů mezi firmami Microsoft a Sun Microsystems.) J# neobsahuje pozdější rozšíření jazyka Java, jako je příkaz assert. V jazyce J# můžeme pochopitelně využívat knihoven prostředí .NET. Vedle toho jsou ale k dispozici i tradiční balíky tříd z knihovny Javy.

Snad nejvíce napoví krátká ukázka programu, něco jako tradiční "Hello, world".

```
package ŽlutáKonina;
import System.*;
import System.Runtime.InteropServices.*;
/** Souhrnný popis třídy ŽlutáKonina.ŽlutoučkovýKůň. */
public class ŽlutoučkovýKůň
{ static String začátek = "Žlutoučkový kůň";
  static String konec = "příšerně úpěl d'ábel ské ódy";
    /** @attributeDllImport("user32", CharSet=CharSet.Ansi) */
    public static native int MessageBox(int hwnd,
    /** @attribute MarshalAs(UnmanagedType.AnsiBStr)*/
    System.String title, System.String caption,
    int type);
    public static void main(String[] args)
    {
      MessageBox(0, konec, začátek, 0);
      Console.WriteLine(záčátek);
      java.lang.System.out.println(konec);
    }
}
```

Příklad ukazuje použití tříd prostředí .NET a použití tříd z JDK pro týž účel. Všimněte si, že J# podporuje dokumentační komentáře. Odpovídající HTML stránky s dokumentací lze v IDE vytvořit příkazem z nabídky.

Podobně jako ostatní jazyky určené pro .NET Framework i Visual J# umožňuje připojovat ke třídám, proměnným, metodám a dalším součástem programu tzv. atributy. Používá se k tomu dokumentační komentář se značkou @attribute. (Připomeňme, že jde o deklarativně zadávané doplňující informace - např. o tom, že daná metoda je naprogramována v dynamické knihovně v neřízeném kódu atd.)

V našem příkladu vidíte výstup na konzolu pomocí objektu System.Consolez knihovny .NET a pomocí javovského objektu java.lang.System.out. V prvním případě text vystoupí ve správném kódování češtiny, zatímco ve druhém (jak je v Javě obvyklé) bychom museli kódování předepsat. Text v dialogovém okně vystoupí správně.

Příklad také ukazuje, že J# dovoluje používat znaky národních abeced v identifikátorech. Jsem si vědom, že po zhlédnutí výše uvedeného příkladu naskáče některým "opravdovým profesionálům" kopřivka, nicméně v posledních letech tuto možnost nabízí stále více programovacích nástrojů, neboť neangličtí programátoři to začínají vyžadovat. Koneckonců, potřebují přemýšlet o řešeném problému, ne o tom, jak se to řekne anglicky.

Typy projektů

Ve Visual J# lze vytvářet podobné projekty jako v ostatních součástech Visual Studia: konzolové a okenní aplikace, knihovny tříd a řídicích prvků, aplikace ASP .NET, webové služby založené na ASP .NET, služby pro Windows nebo aplikace pro mobilní zařízení. Vytvořená seskupení a třídy mají podobné vlastnosti jako třídy a seskupení (assembly) vytvořená v ostatních součástech Visual Studia.

Konverze Javy na J#

Chceme-li převést existující programy v Javě do prostředí .NET, máme následující možnosti:

Máme-li k dispozici zdrojový text programu, zkusíme ho přeložit překladačem J#. Při dodržení dále uvedených podmínek bychom neměli narazit na žádné problémy.

Máme-li k dispozici pouze soubory .class obsahující bajtový kód, můžeme k převodu použít Visual J# Binary Converter Tool (Jblmp.exe). Tento program umí převést většinu souborů s bajtovým kódem do mezijazyka MSIL a vytvořit z něj seskupení, tedy např. spustitelný soubor nebo dynamickou knihovnu pro prostředí .NET. Program Jblmp.exe se spouští z příkazové řádky. Poradí si s většinou standardních knihoven JDK 1.1.4 a s microsoftskými rozšířeními.

Neporadí si s bajtovým kódem, který používá technologii pro spolupráci Javy a komponent COM. Neporadí si také s applety, které v .NET nemají obdobu, s RMI, RNI, JNI a s kódem, který používá konstrukce, jež nebyly v JDK 1.1.4.

Z balíku java.netz Visual J++ jsou podporovány pouze protokoly FTP, HTTP a souborové protokoly, nikoli SMTP, NNTP, a MailTo. Nejsou podporovány také některé třídy z balíku com.ms.com.

Provedení

Microsoft Visual J# .NET 2003 se dodává buď samostatně v provedení Standard, nebo jako součást některého z provedení Visual Studia .NET. (Visual Studio .NET v provedení Standard neexistuje.)

Základní prostředky pro vývoj aplikací v tomto jazyce, tedy překladač spouštěný z příkazové řádky spolu s balíkem dalších nástrojů, lze získat zdarma na internetu v tzv. Microsoft Visual J# .NET redistributable package.

K čemu to?

Na první pohled by se mohlo zdát, že Visual J# .NET je totéž co např. Visual C# .NET: prostě další jazyk, který navíc nepřináší nic nového. Zamyslíme-li se ale nad tímto produktem hlouběji, zjistíme, že jde o velice mocný nástroj v konkurenčním boji s Javou. Nejenže umožňuje programátorům snadno převést existující aplikace z Javy do prostředí .NET, ale navíc umožňuje začít rovnou programovat, aniž by se museli učit nový jazyk a seznamovat se s novými knihovnami. Je to dobrý nástroj, i když by byl ještě daleko lepší, kdyby si uměl poradit s JDK 1.4.

Miroslav Virius