

Silná káva

Nestačí jen přivést dítě na svět, je třeba se o ně i dále starat. Příkladnou rodičovskou péčí o jazyk Java je nutno přiznat firmě Sun Microsystems, která začátkem září 2001 představila svůj další nástroj pro vývoj javovských aplikací: Forte for Java 3.

Forte for Java (FFJ) je produktem pražské softwarové firmy NetBeans, kterou nyní zcela vlastní Sun Microsystems. Program existuje ve třech provedeních, to základní se však nejmenuje Forte for Java, ale NetBeans, je k dispozici zdarma včetně zdrojového kódu a lze si ho stáhnout z internetu. Obsahuje integrované vývojové prostředí včetně vizuálních vývojových nástrojů, nástroje pro ladění, nástroje pro vytváření distribuovaných aplikací podle standardu CORBA a další.

Rozšířené provedení, označené Forte for Java Community Edition, je také zdarma; za svůj vznik údajně vděčí tomu, že někteří investoři nechtějí své podnikání zakládat na produktech typu open source. Jeho vývojové prostředí je prakticky totožné s prostředím NetBeans (navíc jsou tu především nástroje pro tvorbu internetových aplikací a databázový server PointBase), avšak výrobce pro ně zaručuje veškerou podporu. Je samozřejmě také vhodné pro výuku programování v Javě.

Teprve za provedení Enterprise Edition se platí. (V materiálech firmy Sun se říká, že FFJ je i v tomto provedení zdarma a platí se pouze za přidané funkce, jako je modul pro EJB.) Je určeno pro tvorbu podnikových aplikací a k předchozímu přidává WebServices, J2EE a nástroje pro tvorbu Enterprise JavaBeans.

Tento článek vychází ze zkušeností s 60denní zkušební verzí Forte for Java 3 Enterprise Edition.

Vývojové prostředí

Integrované vývojové prostředí (IDE) představuje vlastně uživatelské rozhraní vývojového nástroje a jeho kvality často rozhodují o použitelnosti celého produktu. IDE FFJ se skládá z řady oken, která lze zobrazovat buď jako aplikaci s rozhraním SDI (jednotlivá "volná" okna) nebo MDI (jedno hlavní okno v sobě obsahuje všechna ostatní). Hlavní okno obsahuje nabídku, paletu s vizuálními a nevizuálními komponentami JavaBeans, nástrojový panel atd. Další okna obsahují editor zdrojového kódu, vizuální návrh aplikace, inspektor komponent (okno, v němž lze nastavovat vlastnosti komponent), nástroje pro navigaci v projektu atd.

Editor zdrojového kódu pochopitelně zvýrazňuje syntaxi, a to nejen v Javě, ale i v IDL, XML, HTML a v dalších jazycích, s nimiž se lze v projektech ve FFJ setkat. Barevně také odlišuje části kódu, které lze editovat, a automaticky generované části, které editovat nelze. Mezi drobnosti zpříjemňující práci s tímto editorem patří např. číslování řádků, které usnadňuje orientaci ve zdrojovém kódu. Napíšeme-li v tomto editoru jméno instance nějaké třídy a za ně připojíme tečku, nabídne nám v "bublíně" seznam metod a datových složek, které můžeme použít. Přímou z IDE si lze také prohlížet dokumentaci generovanou pomocí programu javadoc.

Poznamenejme, že základem IDE je víceúčelové jádro, které se mj. stará o správu souborů a oken. K tomuto jádru lze přes otevřené aplikační rozhraní (open APIs) podle potřeby připojovat další moduly. Standardně je takto připojen editor, překladač Javy, ladicí modul, nástroje pro práci s JSP, program javadoc, program pro správu verzí (CVS) a další. Lze tak připojit i uživatelsky vytvořené moduly; existuje prý např. modul (vyvinutý jako studentská práce), který umožňuje pracovat v tomto IDE s jazykem C++. Aplikační rozhraní umožňuje integrovat do IDE mj. i populární editor Emacs, známý především z prostředí Linuxu.

Toto jádro lze využít samostatně a vytvořit si tak vlastní IDE; lze je ale využít i k úplně jiným účelům, třeba i jako aplikační platformu - tedy jako základ vlastních aplikací. (Kdosi nad ním např. vybudoval nástroj CASE založený na modelovacím jazyku UML.)

Vizuální a nevizuální vývoj

Princip vizuálního vývoje aplikací pomocí komponent JavaBeans ve FFJ je v podstatě stejný jako ve většině podobných nástrojů. To znamená, že na paletě v hlavním okně vybereme myší požadovanou komponentu a klepnutím ji vložíme do vizuálního návrhu okna. Vlastnosti komponent (barvu, polohu, velikost, písmo a další) pak můžeme upravit v okně inspektoru komponent.

IDE FFJ ihned generuje odpovídající zdrojový kód. To znamená, že neukládá kroky vizuálního vývoje do pomocných souborů, z nichž by se teprve později - např. při uložení nebo před překladem -

vytvářel kód. (Tak to dělá např. PowerJ nebo VisualAge for Java.) IDE zobrazuje existující kód, a to celý - což bohužel o některých konkurenčních nástrojích říci nelze. Při práci s událostmi vytváří IDE kostry handlerů (obslužných metod) a také registruje příjemce událostí.

IDE také umožňuje automaticky generovat nejčastější operace. K tomu používá tzv. propojení: Vybereme komponentu a v hlavním okně stiskneme tlačítko, kterým vyvoláme propojovacího šamana (Connection Wizard). Ten se pomocí několika dialogových oken vyptá na vše potřebné a pak vytvoří odpovídající kód. Tak lze např. bez napsání jediného řádku kódu naprogramovat, že po stisknutí tlačítka se z textového pole, tj. z komponenty JTextField, přenesou napsaný text do seznamu, tj. do komponenty JList. (S podobným nástrojem je možné se setkat ve VisualAge for Java od IBM, které ale při generování kódu využívá pomocného souboru.)

Už bylo řečeno, že IDE barevně odlišuje automaticky generované části kódu, které nelze editovat, a části, které jsme napsali "ručně" a které editovat lze. Chceme-li automaticky generované části kódu změnit nebo odstranit, musíme k tomu použít vizuálních nástrojů. FFJ tedy není dvoucestný vývojový nástroj.

Generovaný kód

Zdrojový kód se tedy při vizuálním programování v IDE generuje ihned a editor jej celý zobrazuje. Vytvořený kód je založen na běžných návrhových vzorech pro Javu 2. To znamená, že např. o ukončení aplikace při uzavření hlavního okna se postará metoda anonymní instance anonymní vnitřní třídy, která je potomkem standardní třídy WindowAdapter. (Pokud vám to připadá samozřejmé, pak vězte, že např. JBuilder - i když využívá JDK 1.3 a knihovnu Swing - generuje kód z JDK 1.0: nejprve povolí vznik událostí v okně a pak předefinuje chráněnou metodu processWindowEvent(). V PowerJ zase třída okna implementuje rozhraní WindowListener a registruje se jako příjemce událostí. V důsledku toho obsahuje třída okna zpravidla řadu prázdných metod uvedeného rozhraní.)

Obsluha běžných událostí - např. stisknutí tlačítka - je ve FFJ opět založena na anonymních vnitřních třídách, jejichž metody volají skutečný handler (typicky metodu okna). Ani tento postup není úplně samozřejmý, neboť např. v PowerJ implementuje třída okna potřebné rozhraní, např. ActionListener, a okno se registruje jako příjemce událostí.

České znaky s diakritickými znaménky (např. nápisy na komponentách) se do zdrojového kódu vždy ukládají v podobě řídicích sekvencí UNICODE, což není příliš přehledné, zaručuje to však kompatibilitu při přechodu do jiných prostředí.

Doplňme, že vytvořený kód je poměrně velmi přehledný; nejspíš jde o výsledek tlaku daného tím, že IDE kód zveřejňuje ihned a celý.

Java

FFJ pochopitelně podporuje JDK 1.2. Zajímavé ale je, že JDK není součástí instalace, musíme si je instalovat zvlášť. Pro instalaci IDE je nezbytné mít na cílovém počítači nejprve instalaci Java 2 SDK verze 1.3.1.

Pokud máme na cílovém počítači několik instalací JDK, můžeme si ve FFJ kdykoli předepsat, kterou z nich chceme při vývoji používat. Vytvořené aplikace bez problémů zvládají češtinu; není tedy třeba upravovat soubor font.properties.

Databáze

Podobně jako jiné vizuální vývojové nástroje založené na jazyce Java, i FFJ obsahuje prostředky, které usnadňují vývoj databázových aplikací. Jejich základem jsou především komponenty využívající JDBC, a to vizuální a nevizuální; najdeme je na stránce JDBC palety komponent.

Vizuální databázové komponenty jsou "obyčejné" komponenty Swing (JCheckBox, JTable, JList aj.), které prostřednictvím připojeného datového modelu zobrazují informace z databázových tabulek a umožňují s nimi manipulovat.

Nevizuální databázové komponenty jsou komponenty JavaBeans, které zacházejí s daty uloženými v databázích. Představují připojení k databázi, sdílené připojení (pooled connection), uložené procedury, databázový navigátor, různé varianty datové množiny (rowset) a další. Poznamenejme ještě, že jedna z variant této komponenty umožňuje import a export dat ve formátu XML.

Pro vytváření oken databázových aplikací ve FFJ pomáhá šaman (JDBC Form Wizard), který se postará o navázání spojení, zprostředkuje volbu databázových tabulek a pohledů, určí úroveň izolace v transakcích, zeptá se, které sloupce z tabulek chceme zobrazovat atd., a na základě toho vytvoří odpovídající kód.

Vedle toho je ve FFJ k dispozici transparentní perzistence (Transparent Persistence); za tímto označením se skrývá možnost zobrazovat data v relačních databázích dostupných pomocí JDBC jako

objekty jazyka Java a tak s nimi i manipulovat, aniž bychom museli znát SQL, JDBC API atd. Pomocí transparentní perzistence se jednotlivé sloupce z tabulky zobrazí do javovských objektů předdefinovaných tříd. Vztahy mezi daty v různých tabulkách založené na vnějších klíčích se převedou na vztahy mezi objekty založené na referencích nebo vytvořené uložením v jednom kontejneru.

Dodejme, že součástí instalace FFJ (už i v Community Edition) je relační databázový server PointBase Network Edition 3.5 spolu s administračními nástroji. Tento server lze použít především pro účely vývoje a testování databázových aplikací.

CORBA

Podpora tvorby distribuovaných aplikací podle standardu CORBA je součástí všech verzí FFJ včetně NetBeans. Jak je obvyklé, obstarají většinu práce šamani, kteří vytvoří potřebný kód. Vedle toho IDE využívá služeb objektové sběrnice (ORB) k připojení ke jmenné službě a k Interface Repository. Implicitně pracuje s implementací OpenORB, lze ho však přepojit na libovolnou implementaci vyhovující standardu OMG CORBA 2.3.

Vzhledem k tomu, že různé implementace objektové sběrnice nejsou vždy navzájem zcela kompatibilní, nabízí FFJ kompatibilitu se 13 nejčastěji používanými; z nich uvedme alespoň (v abecedním pořadí) CORBA J2EE, JavaORB 2.2.x, JDK 1.2 a JDK 1.3, Orbix, VisiBroker 3.4 a 4.x.

Enterprise JavaBeans

Na komponenty Enterprise JavaBeans (EJB) se můžeme dívat jako na panely, z nichž jsou budovány dnešní vícevrstvé podnikové aplikace. S trochou zjednodušení můžeme říci, že to jsou komponenty JavaBeans, které implementují jistá předepsaná rozhraní. Obsahují aplikační nebo prezentační logiku, starají se o spolupráci s databázemi, servery, o přístup ke klientům atd. Komponenty EJB lze používat jedině tak, že je spolu s dalšími souvisejícími komponentami uložíme do modulu EJB a umístíme v tzv. aplikačním serveru.

Typický aplikační server funguje jako kontejner pro EJB, zprostředkovává komunikaci prostřednictvím standardu CORBA a funguje také jako webový server s podporou Javy (tj. umožňuje i běh servletů a JSP). Příkladem takového serveru je iPlanet od firmy Sun Microsystems, který ale není součástí dodávky FFJ. (Pro vývoj a ladění EJB ovšem není aplikační server nezbytný.)

Součástí FFJ Enterprise Edition je i referenční implementace EJB pro J2EE, což je kontejner pro EJB, který pro ladění EJB plně dostačuje. Vedle toho ovšem obsahuje IDE plug-in pro server iPlanet, takže pokud je k dispozici, můžeme jej využít.

Aplikace pro web

Mezi aplikacemi pro web se dnes těší asi největší popularitě servlety a stránky JSP. Je tedy pochopitelné, že FFJ podporuje jejich vývoj (včetně např. vytváření knihoven uživatelských značek). Součástí instalace je webový server Apache Tomcat 3.2, který lze používat buď jako malý samostatný webový server, nebo jako doplněk k webovému serveru Apache. Při ladění servletů a JSP ve FFJ jej lze využívat přímo z IDE.

Ve skutečnosti se servlety, stránky JSP a jiné webové komponenty (javovské komponenty běžící na straně webového serveru) sdružují do webových aplikací (modulů). FFJ umí vytvořit nezbytnou strukturu webové aplikace a také zabalit vytvořené a přeložené třídy jako archiv ve formátu WAR (Web Archive) spolu s deskriptorem ve formátu XML. Jistě nepřekvapí, že pro všechny tyto operace jsou v IDE připraveni šamani.

Součástí Enterprise Edition je také modul zvaný Web Services. Představuje nástroj pro vývoj webových služeb založených na protokolu SOAP a jazyku WSDL, který umožňuje dát vlastní komponenty s aplikační logikou k dispozici prostřednictvím WWW. Vývoj webových služeb ve FFJ je založen na šamanech, kteří se postarají o vše potřebné včetně vytvoření potřebné infrastruktury, takže není potřeba psát prakticky žádný kód.

Jak známo, webové služby založené na SOAP využívají komunikace prostřednictvím jazyka XML a protokolu http, který firewally bezpečně propouštějí. Fungování webové služby vyvinuté ve FFJ lze ve stručnosti popsat takto: Když webová služba přijme vstupní dokument, tj. klientský požadavek ve formě dokumentu XML, předá jej odpovídající metodě. Tato metoda analyzuje dokument a převede jeho prvky na parametry metod, které dále zavolá. Výsledky, které tyto metody vrátí, převede do jazyka XML, tj. vytvoří výstupní dokument a ten odešle klientovi. Webové služby typicky spolupracují s komponentami EJB umístěnými ve vhodném aplikačním serveru a jejich prostřednictvím pak využívají dat v databázích.

Systémové požadavky

Už víme, že chceme-li instalovat FFJ Enterprise Edition, musíme si nejprve samostatně instalovat

JDK 1.3.1, Standard Edition. Na instalačním CD instalaci JDK samozřejmě najdeme; při instalaci z internetu si jej musíme stáhnout zvlášť.

FFJ lze instalovat v prostředí operačních systémů Linux Red Hat 6.2+, Mac OS X, OS/2, HP-UX, OpenVMS, Solaris 8 for Sparc, Windows 98 a vyšších. Pro Windows NT 4.0 je nutný Service Pack 6, pro Windows 2000 je požadován Service Pack 2.

Minimální hardwarové požadavky představuje počítač s procesorem Pentium III/866 MHz a 384 MB RAM, pro Linux též počítač s 512 MB RAM a se swapovacím prostorem 128 MB. Dodejme ale, že při takovéto konfiguraci už lze FFJ opravdu používat. Na disku zabere instalace ve všech případech cca 110 MB.

Proces instalace FFJ je bez problémů; pokud jsme na svém počítači měli nějakou předchozí verzi tohoto produktu, může instalační program převzít její nastavení.

Dojem

Ve srovnání s konkurenčními produkty, s nimiž se lze setkat na našem trhu, může být předností Forte for Java nabídka webových služeb v provedení Enterprise, podpora standardu CORBA ve všech provedeních, ale také snadné ovládání a přehlednost generovaného kódu. Možná vám to bude připadat nepodstatné, nicméně za velkou přednost také považuji, že mi prostředí ukazuje vždy všechny vygenerovaný kód a nic přede mnou neskrývá.

Za nevýhodu FFJ lze považovat značné nároky vývojového prostředí a skutečnost, že nejde o dvoucestný nástroj (nelze modifikovat automaticky generovaný kód). Při rozhodování o koupi může svou roli hrát i skutečnost, že mnohé konkurenční produkty obsahují jako součást provedení Enterprise také aplikační server - i když jen vývojářskou licenci. Nic to však nemění na faktu, že Forte for Java 3 Enterprise Edition je bezpochyby velice silný a kvalitní produkt.

Miroslav Virius

Forte for Java 3 Enterprise Edition

Nástroj pro vývoj javovských aplikací pod Windows 98 a vyššími, Linux Red Hat 6.2+, Mac OS X, OS/2, HP-UX, OpenVMS a Solaris 8 for Sparc.

Hardwarové nároky: PC s Pentiem III/866 MHz a 384 MB RAM, (pro Linux 512 MB RAM a 128 MB swapovacího prostoru), 110 MB na disku.

Výrobce: Sun Microsystems Inc., USA (www.sun.com).

Poskytl: Sun Microsystems Czech, Praha (www.sun.cz).

Orientační cena: licence 2060 USD, média 26 USD.