

STARÝ PES S NOVÝMI KOUSKY?

Z historie programovacích jazyků

Jedním z kritérií, kterými lze rozdělit programovací jazyky, je jejich konstrukce. Zda jsou konstruovány jako procedurální nebo objektivě orientované.

Jako procedurální jsou označovány jazyky, v nichž je výsledný kód uspořádán do shluků úloh, které se mají v rámci daného zadání vykonat. Typickým představitelem procedurálních jazyků je FORTRAN, jeden z nejstarších programovacích jazyků vůbec.

Do druhé, z historického pohledu mladší, větve náleží jazyky založené na konceptu objektů. Kód napsaný objektivě orientovanými jazyky je vykonáván zpravidla na základě událostí, které mohou být vyvolány libovolně, bez nutnosti zachovávat určitý sled.

My se v tomto článku nedbeme detailně zaobírat popisem jednotlivých skupin jazyků, ale zaměříme se na jednu vývojovou větev, která má svůj počátek v roce 1966. Tehdy vznikl procedurální jazyk **MUMPS** (*Massachusetts General Hospital Utility Multi-Programming System*), jenž umožňoval multitasking (byl provozován na Unixu), zahrnoval hierarchickou databázi a byl to interpret.

Díky uvedeným charakteristikám se brzy rozšířil a roku 1977 byl standardizován jako **ANSI M**. Vzniklo několik implementací, z nichž nejvýznamnější byly GTM, DSM, DTM, ISM, MSM. Za každou z stála jiná firma. Byl portován také pro VMS OS a Windows, které byly okrajovou záležitostí.

Až do příchodu PC tvořily implementace M jazyka základní informační pilíře společností na Západě i v zemích bývalého Sovětského bloku. Mezitím zaznamenaly prudký rozvoj relační databáze a během 90. let začaly vytěšňovat jiné aplikace včetně M typů.

Jak z této situace ven?

Okolo vás existují relační databáze, objektivě jazyky a vy máte superrychlý jazyk

pracující se stromovými strukturami, neznalý objektů, relačních tabulek. Co se nabízí? Přidat podporu SQL a naučit jazyk řeči objektů.

M jako takový končí rokem 1997, kdy firma InterSystems uvedla nový produkt, objektovou databázi a aplikační server **Caché**, jenž používá jazyk ObjectScript, který má syntakticky mnoho společných příkazů s M, ale obsahuje aparát pro práci s objekty. Aplikace napsaná v M pobeží víceméně bez změny zdrojového kódu též v Caché.

Tím zůstala původním aplikacím otevřená cesta k budoucnosti. M aplikace se jednoduše vezme, nasadí na Caché server a pracuje stejně jako dosud. Navíc, díky novým technologiím zabudovaným do Caché, je možno takové aplikace restrukturalizovat do vícevrstvé architektury a provozovat pod libovolným uživatelským rozhraním (tlustým klientem, webovým klientem).

Technické pozadí řešení

Příkazy ObjectScriptu dovedou manipulovat s daty v databázi Caché i pracovat se zařízeními operačního systému, ať se již jedná o soubory, či pásková zařízení, fronty či samotný shell.

Každý příkaz je ve skutečnosti funkcí napsanou v C. Tedy je jednoduché napsat mocné příkazy, které budou ale ze syntaktického pohledu velmi stručnými. V tom spočívá hlavní síla M a potažmo ObjectScriptu – možnost psát příkazy jedno- či dvouřádkově (samozřejmě i celými slovy) a docílit velké změny. Většina jiných jazyků je v tomto ohledu poněkud upovídaná (Java či .NET).

Samozřejmě že mladší generace programátorů dává přednost používání objektivě syntaxe. Z toho důvodu byla pro práci se soubory vytvořena sada tříd, které v metodách používají původní příkazy ObjectScriptu. Ale jelikož správné objekty

Vývoj programovacích jazyků

oreilly.com/news/graphics/prog_lang_poster.pdf

odstiňují těla metod od okolí, programátor pracuje s metodami Read, ReadLine, Write, WriteLine a je nepodstatné, že tyto metody používají příkazy ObjectScriptu open, read, write, close a jiné. Takových sad (balíčků) tříd je v Caché několik desítek, a pokrývají nejrůznější oblasti vývoje.

V Caché lze definovat i uživatelské třídy různých typů. Metody Caché tříd mohou být implementovány jazykem ObjectScript. Třídám, které jsou označeny jako persistentní, se automaticky generuje SQL projekce do jedné či více relačních tabulek, podle složitosti třídy. Díky tomu je možno pracovat s daty prostřednictvím objektivě notace či SQL příkazů. Tak jako tak, za běhu se vždy vykonává sada příkazů ObjectScriptu generovaná serverem automaticky v okamžiku kompilace třídy. Ovšem uživatel je opět od nízkoúrovňového kódu odstíněn.

Pro úplnost dodejme, že data jsou stále ukládána v hierarchických (stromům podobných) strukturách, jen jsou viditelná díky různým větvím generovaného kódu různým způsobem – objektivě, relačně, nebo i jako XML dokumenty.

Díky novému jádru, vycházejícímu z toho dobrého, co implementoval původní jazyk ANSI M, a podporujícímu objektivě notaci, zažívá v základě 40 let stará technologie svou renesanci. Dnes lze Caché provozovat na téměř každém operačním systému počínaje Unixy, Windowsy, Linuxy, OpenVMs až po Mac OSX. Instalovaná báze se blíží 150 000 systémů s asi 5 000 000 uživateli. Na Caché jsou provozovány všechny možné druhy databázových aplikací, počínaje zdravotnickými informačními systémy přes bankovní systémy, ERP aplikace a aplikace pokrývající potřeby mnoha jiných oborů. □

