

■ Relácie

eq – rovná sa
neq – nerovná sa
gt – väčší ako
lt – menší ako
geq – väčší alebo rovný ako

leq – menší alebo rovný ako
equivalent – ekvivalencia
approx – aproximácia
factorof – deliteľ

Príklad:

```
<apply>
<equivalent/>
<ci> a </ci>
<apply>
<not/>
<apply> <not/> <ci> a </ci> </apply>
</apply>
</apply>
```

■ Výpočty a vektorové výpočty

int – integrál
diff – diferenciacia
partialdiff – parciálna diferenciacia
lowlimit – dolná hranica integrálu
uplimit – horná hranica integrálu

bvar – ohraničená premenná
degree – stupeň operátora
divergence – divergencia
grad – gradient

Príklad:

```
<apply>
<int/>
<bvar><ci> x </ci></bvar>
<lowlimit><ci> a </ci></lowlimit>
<uplimit><ci> b </ci></uplimit>
<apply>
<ci type="function"> f </ci>
<ci> x </ci>
</apply>
</apply>
```

■ Elementárne funkcie

sin, cos, tan, sec, csc, cot, sinh, cosh, tanh, sech, csch, coth, arcsin, arccos, arctan, arccosh, arccot, arccoth, arccsc, arccsch, arcsec, arcsech, arcsinh, arctanh – trigonometrické funkcie

exp – e^x
log – logaritmus
ln – prirodzený logaritmus

Príklad:

```
<apply>
<exp/>
<ci> x </ci>
</apply>
```

■ Štatistika

mean – stredná hodnota
sdev – smerodajná odchýlka
variance – odchýlka
median – median

mode – modus
moment – moment
momentabout – bod momentu

Príklad:

```
<apply>
<mode/>
<ci> x </ci>
</apply>
```

■ Lineárna algebra

vector – vektor
matrix – matica
matrixrow – riadok matice
determinant – determinant

tnaspose – transponovanie
vectorprodukt – vektorový súčin
scalarproduct – skalárny súčin

Príklad:

```
<apply>
<transpose/>
<ci type="matrix"> A </ci>
</apply>
```

■ Teória množín

set – množina
list – zoznam
union – zjednotenie
intersect – prienik
in – náležanie
notin – negácia náležania
subset – podmnožina

notsubset – negácia podmnožiny
setdiff – množinový rozdiel
card – mohutnosť množiny
cartesianproduct – karteziánsky súčin

Príklad:

```
<apply><cartesianproduct/>
<ci> A </ci>
<ci> B </ci>
</apply>
</apply>
```

■ Postupnosti a rady

sum – suma
product – produkt

limit – limita

Príklad:

```
<apply>
<product/>
<bvar><ci> x </ci></bvar>
<condition>
<apply> <in/>
<ci> x </ci>
<ci type="set"> B </ci>
</apply>
</condition>
<apply><ci type="function"> f </ci>
<ci> x </ci>
</apply>
</apply>
```

■ IGOR KULMAN

S Caché v štúdiu IX.

Java je jazyk mnohými zatracovaný, mnohými vyzdvihovalý. Názory sa rôznia. Pravdou však ostáva, že Java je jedna z možností, ako jednoducho vytvárať portabilné aplikácie. Väčšina aplikácií je databázová, a tak nie je nič zvláštne, že o Javu sa zaujímajú aj autori databázových platforiem.

Základom prepojenia klientských aplikácií v prostredí Caché a Javy je knižnica CacheDB.jar. Táto knižnica sa s príchodom verzie Caché 5 dočkala prepracovania a úplnej optimalizácie pre prostredie J2EE.

Triedu definovanú v Caché možno klientovi v Jave sprístupniť veľmi jednoducho – stačí opísať jej Java projekciu. Pri preklade takejto triedy prekladač sám automaticky vygeneruje čistý kód Javy a uloží ho do jedného alebo viacerých súborov. Počet súborov sa odvíja od počtu referenčných atribútov danej triedy. Celý proces je kompletne automatizovaný,



používatelia nemusia pamätať na vytváranie projekcii odkazovaných tried – stačí iba dané súbory java vložiť do príslušného projektu alebo na ne odkázať z klientských tried Java. V rámci definície triedy v Caché možno pri jednotlivých vlastnostiach navyše určiť ich Java dátový typ.

Princíp objektového prepojenia tried Caché a tried Java je podobný ako pri technológii COM. Aj tu sa vždy na strane klienta otvára lokálna kópia inštalácie Caché triedy a pracuje sa s lokálnou kópiou. Tá je podľa potreby synchronizovaná s inštaláciou na serveri. Na rozdiel od možnosti využívať Visual Basic však Caché neponúka žiadneho sprievodcu návrhom formulára.

Pripojenie klienta k serveru sa môže vykonať metódou ConnectToCache, napríklad s takýmto obsahom:

```
public boolean ConnectToCache(String url) {
    try {
        cacheConnection =
        CacheDatabase.getDatabase(url);
        return true;
    } catch (Exception ex) {
        System.out.println("Chyba: "
        + ex.getClassName()
        + ": " + ex.getMessage());
        ex.printStackTrace();
        return false;
    }
}
```

Spojenie s Caché je udržiavané inštanciou cacheConnection triedy Database.

Len čo je spojenie raz vytvorené a formulár otvorený, možno pracovať s lokálnou klientskou inštanciou danej triedy. Caché implementuje spojenie do Javy dvojakým spôsobom – plnohodnotne a „odľahčené“, čo je podstatne rýchlejšie, ale má obmedzený rozsah funkčnosti. Je na vývojarovi, aby si vybral, ktoré z týchto pripojení použije.

Medzi ďalšie možnosti pripojenia patrí využitie rozhrania JDBC alebo technológie Enterprise Java Beans. Caché využíva v prípade EJV výkonné BMP beans s tým, že všetok kód na správu je generovaný automaticky. Firma InterSystems sa zamerala na integráciu Caché EJB s produktom firmy BEA WebLogic. Projekciu možno vytvoriť aj pre ďalšie servery, tie však nie sú oficiálne podporované.

■ MAREK KOČAN