

Caché 5.1 - Pokročilé typy vlastností tried Caché, kolekcie a ich projekcie do Javy

Tentoraz si ukážeme prácu s pokročilými typmi vlastností tried Caché a kolekciami a ich projekciou do Javy.

Na to si upravíme v Caché Studiu triedu **ucto.demo**.

Dennik; pridáme nasledujúce dve vlastnosti:

```
/// zoznam oprávnených osôb, ktoré smú robiť
úctovné zápisy v danej úč. jednotke
Property opravnenuzivatelia As list Of
$String(JAVATYPE = "java.util.List");

/// rôzne dokumenty vzťahujúce sa na danú
úctovnú jednotku
Property zivnostenskylist As $Stream.Global-
Binary;
```

Property poznamka As \$GlobalCharacterStream;

Ďalej ešte pridáme definíciu projekcie:

```
Projection Java As $Projection.Java(ROOTDIR =
"C:\Software\Projects\Java\netbeans-pro-
jects\mojeucto\src\mojeucto");
```

Caché podporuje dva typy viachodnotových vlastností, a to pole (asociatívne) a zoznamy. Pri generovaní projekcie do Javy Caché automaticky prevádza tieto štruktúry do štandardných Java tried **java.util.Map**, respektíve **java.util.List**. My sme pri vlastnosti **opravnenuzivatelia** uviedli explicitne prostredníctvom parametra **JAVATYPE**, že zoznam Caché má byť vystavený v Jave ako **java.util.List**. Toto v danom prípade nie je potrebné, generátor by to urobil automaticky, ale berie to ako ukážku toho, ako možno používateľsky ovplyvniť dátové typy vlastností proxy tried.

Výsledný vygenerovaný zdrojový kód vlastnosti **opravnenuzivatelia** v Jave teda vyzerá takto:

```
public java.util.List getopravnenuzivatelia() throws com.intersys.objects.Cache-
Exception {
    .. implementace
    return (java.util.List) (cobj.new-
JavaInstance());
}
```

Pre vlastnosť **zivnostenskylist** sa vygenerujú dve metódy, jedna na vstupné a druhá na výstupné operácie:

```
public com.intersys.objects.CacheInputStream
getzivnostenskyListIn() throws com.intersys.objects.CacheException {
    .. implementace
    return (db.getInputStream (cobj));
}

public com.intersys.objects.CacheOutputStream
getzivnostenskyListOut() throws com.intersys.objects.CacheException {
    .. implementace
    return (db.getOutputStream (cobj));
}
```

K vlastnosti **poznamka** sa tiež vygenerujú dve metódy, ale budú vracajú triedy **Reader** alebo **Writer**, pretože poznámka je znakový reťazec.

Ak máme vystavenú Java projekciu triedy **Dennik**, môžeme vložiť nové dáta do databázy hoci takto:

```
///... deklarácia a inicializácia
String url="jdbc:Cache://localhost:56773/"
+ namespace;

/* Connect to this machine, in the SAMPLES
namespace */
dbconnection = CacheDatabase.getDatabase
(url, username, password);

// ! zjednodušenie: máme iba jeden dennik,
s id = 1,
dennik =
(Dennik)Dennik.open(dbconnection,new Id(1));
System.out.println(dennik.getucetniJed-
notka());

// prida noveho opravneneho pouzivatela;
dennik.getopravnenuzivatelia().add(ou);
dennik.save();
System.out.println(dennik.getopravnenuziva-
telia().size());
dbconnection.closeObject(dennik.getoref());
denik = null;

// zmeníme / pridáme poznamku
dennik.getpoznamkaOut().write("Toto je druhy
test!");
dennik.save();
```

```
// podobne i binarne data - pozri uplny pri-
klad

/* Close the connection */
dbconnection.close();
```

Úplný zdrojový kód príkladu nájdete na www.infoware.sk v sekcii **Programujeme**.

Teraz ešte preberieme možnosti používateľského spresnenia spôsobu generovania projekcií do Java tried, sľúbené už minule.

Trieda **%Projection.Java**, ktorá je zodpovedná za tvorbu proxy tried, má niekoľko parametrov, určujúcich správanie procesu projekcie a výsledku.

Základný parameter, s ktorým sme sa už oboznámili, je **ROOTDIR**. Ten určuje kmeňový adresár, v ktorom sa budú tvoriť podadresáre podľa balíčka triedy Caché a kde budú uložené proxy triedy.

Ďalšie parametre sú o. i. tie, ktoré uvádzame v tabuľke:

Názov	Hodnoty	Opis
JAVADOC	1, 0	Určuje, či sa bude automaticky generovať príslušná dokumentácia
PRIMITIVEDATATYPES	1, 0	Určuje, či dátové typy Caché budú v proxy triedach vystavené ako primitívne dátové typy Javy
NEWCOLLECTIONS	1, 0	Tento parameter je užitočný hlavne pri starších aplikáciách, určuje, či sa dátové typy Caché typu zoznam či pole budú v Jave vystavovať ako java.util.List či java.util.Map alebo použijú ones už zahrnuté triedy z knižnice CacheDB.jar
RECURSIVE	1, 0	Určuje, či sa pri zostavovaní proxy triedy majú generovať i projekcie závislých tried
POJO	1, 0	Určuje, či sa vystavené triedy majú generovať ako POJO (klasické obyčajné Java triedy) objekty alebo ako pôvodné proxy triedy
CHECKUPTO-DATE	1, 0	Určuje, či sa projekcie majú generovať vždy alebo len pre triedy, ktoré boli zmenené od poslednej kompilácie

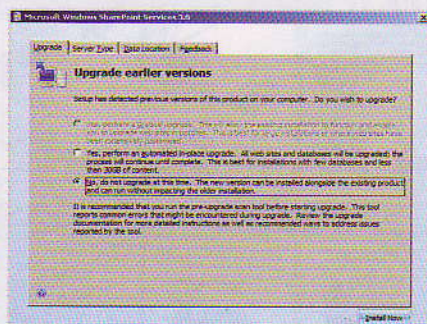
V nasledujúcej časti si ukážeme prácu s vloženými dopytmi tried Caché.

■ DANIEL KUTÁČ,
Senior Sales Engineer spol. InterSystems B.V.
kutac@intersystems.cz

Tipy a triky pre Office

Windows SharePoint Services 3.0 na SBS server 2003

Malé firmy, ktoré využívajú server Windows SBS 2003, majú k dispozícii zaujímavú technológiu Windows SharePoint Services 2.0 (WSS), ktorá je výborným doplnkom k MS Office na tímovú spoluprácu. Priamo z aplikácie MS Office možno ukladať dokumenty do databázy, sledovať históriu zmien, prepojiť kalendár a úlohy MS Outlook s tímovým webom a podobne. Nová verzia SharePoint 3.0 prináša ďalšie vymoženosti, ktoré výrazne zlepšujú prácu s touto technológiou.



Nasadenie verzie 3.0 je jednoduché, ale musí sa dodržať presný postup inštalácie, lebo inak prestanú fungovať nie-

ktoré funkcie SBS servera, založené na WSS 2.0. Preto nemožno urobiť upgrade, ale paralelnú inštaláciu vedľa pôvodnej verzie. Nevyhnutným predpokladom na úspešnú inštaláciu je nainštalovaný Microsoft .Net framework 3.0. Keby v systéme nebol, inštalácia zastaví aj tri služby: Internet Information Services, SharePoint Administration Service, SharePoint Timer Service. Veľmi dôležité je skontrolovať v ďalšom kroku, či je vybratá možnosť novej inštalácie a samostatný server. Keď tu urobíte chybu a budete pokračovať v inštalácii, môžete počítať aj s možnosťou použitia zálohy servera na jeho obnovu do funkčného stavu.

Po nainštalovaní WSS 3 je nevyhnutné urobiť ešte následné konfiguračné úkony. Treba spustiť sprievodcu konfiguráciou, ktorý má 10 krokov a prebieha automaticky. Ak sa nespustí po skončení inštalácie, treba ho vyhľadať v ponuke **Štart > Nástroje na správu > Sprivodca konfigurácie produktov a technológie servera SharePoint**. Po skončení tohto kroku sa zobrazí internetový prehliadač s novou stránkou WSS. Po tomto kroku nefunguje pôvodná verzia WSS 2.0 (<http://companyweb> ani stránka <http://názovservera>). Vo štvrtom kroku vytvárate stránku webovej aplikácie, kde zadávate jej názov, port, adresu, overovanie a vyhľadávací server. To urobíte v centrálnej správe WSS kliknutím na **Správa aplikácie** a potom na **Vytvorenie alebo rozšírenie webovej aplikácie**. Následne treba vytvoriť stránku najvyššej úrovne zadaním jej opisu, adresy a použitím tímovej šablóny. Znovu pokračujete na stránke centrálnej správy.

V piatom kroku treba vymazať preddefinovanú stránku, ktorá sa vytvorila v treťom kroku inštalácie (

servera) spolu s databázou obsahu a IIS webu. Celý tento postup urobíte cez centrálnu správu WSS 3.0. Znovu musíte skontrolovať, či vymazávate správny web. V poslednom kroku treba spustiť web zastavený v Internet Information services <http://názovservera> a <http://companyweb> (ak je takisto zastavený). Tým je nasadenie novej verzie ukončené a môžete používať aj doterajšiu verziu, aj novú. Ak ste dodržali všetky pokyny, mali by normálne fungovať aj všetky funkcie servera založené na WSS 2. Pred inštaláciou sa odporúča zálohovať server a dôkladne si preštudovať aktuálne poznámky a podrobné pokyny na inštaláciu na SBS 2003, ktoré sú dostupné na internetovej stránke Microsoftu.

■ PETER BELKO,
belko.peter@gmail.com