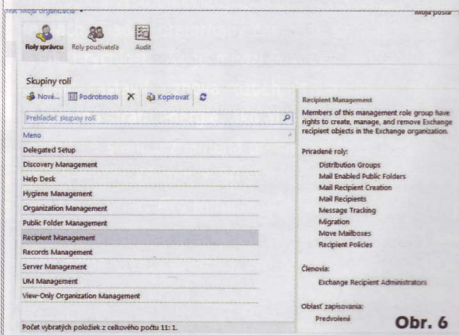


rules) a môže vyhľadávať správy o doručení v používateľských poštových schránkach. V prípade, že je správcom priradená rola správca Discovery Management, pribudne v sekcii Ovládací prvok



Obr. 6

pošty možnosť Zisťovanie, ktorá umožňuje zadávať požiadavky na vyhľadávanie správ vo viacerých poštových schránkach. Výsledky hľadania sa ukladajú do špeciálnej poštovej schránky, označenej Discovery Search Mailbox.

V poslednej opísanej sekcii Telefón a hlas nájdete správca v podskupine Prístup k synchronizácii ActiveSync možnosť definovať pravidlá na prístup pre špecifické série alebo modely pripájaných zariadení. Takisto možno definovať nastavenie prístupu pre zariadenia, ktoré nie sú spravované pravidlami ani osobnými výnimkami. Nastavenie prístupu pre zariadenia rozoznáva tri možnosti: Povolíť, Blokovat alebo Karanténa. Správca môže spravovať aj zariadenia umiestnené

do karantény a rozhodovať o ich povolení alebo blokovaní. Ďalej môže vytvárať, odstraňovať a upravovať politiky aplikované na zariadenia synchronizované technológiou Exchange ActiveSync.

Nakoniec má správca pri výbere kontextu Iný používateľ... možnosť vybrať iného používateľa, ktorého nastavenia treba modifikovať. V tomto prípade má k dispozícii iba podskupinu nastavení opísaných v kapitole Možnosti pre používateľa.

Exchange Control Panel môže obsahovať rôzne kombinácie možností v závislosti od delegovaných oprávnení používateľa, preto sa v uvedenom prehľade nevyskytujú úplne všetky.

MAROŠ FÜLEKÝ, CIT

Tipy a triky s Caché Try – catch ošetrenie chýb a prepojenie do systémového protokolu chýb

Na početné otázky priaznivcov programovania s vysokovýkonnou objektovou databázou InterSystems Caché vám prinášame niekoľko aktuálnych tipov a trikov.

Tento tip súvisí so spoločným ošetrovaním chýb, ktoré si môžete pripomenúť na odkaze <http://www.intersystems.cz/education/university/serialy/lesson/T&T%2015.htm>.

Od čias publikovania nášho seriálu o Caché uplynulo už dosť času a odvtedy sa v tejto objektivej databáze veľa zmenilo. Hlavnou zmenou v oblasti práce s výnimkami bolo zavedenie mechanizmu try – catch vo verzii 2007.1. Ak teda chceme v novom kóde ošetriť dynamickú výnimku, môžeme ľahko napísať hoc aj toto:

```
podil(a,b) public
{
    #dim ex as %Exception.AbstractException
    s res=0
    try {
        s res=a/b
    } catch (ex) {
        w ex.DisplayString()
    }
    q res
}
```

Toto je trápne jednoduchá demonštrácia nového spôsobu zachytávania výnimiek. Všimnite si, že po novom je výnimka objektom, a nie iba premennou. Ak chceme sami vyvolať výnimku umelo, použijeme príkaz **throw ex**.

Veľmi často sa asi stretnete s potrebou dať do súladu systémový objekt **% Status**, ktorý triedy Caché veľmi často využívajú na vrátenie informácie o priebehu spracovania kódu metód. Kým **% Status** je návratová hodnota, ktorá môže obsahovať nejakú informáciu o chybe / chybách, táto informácia sa nepovažuje za runtime výnimku. No **% Status** možno jednoducho na runtime výnimku previesť, a to pomocou metód tried v balíku **% Exception**. Teda môžeme napísať niečo takéto:

```
try {
    s tOsoba##class(tt.osoba).%OpenId(100,4,.tSC)
    if $$System.Status.IsError(tSC) throw ##class(%Exception.StatusException).CreateFromStatus(tSC)
} catch (ex) {
    w ex.DisplayString()
}
q
```

To je pekné, ale my by sme radi použili logovanie chýb do systémového denníka pomocou volania **BACK ^% ETN**. Takže predchádzajúci kód rýchlo upravíme:

```
try {
    s tOsoba##class(tt.osoba).%OpenId(100,4,.tSC)
    if $$System.Status.IsError(tSC) throw ##class(%Exception.StatusException).CreateFromStatus(tSC)
} catch (ex) {
    w ex.DisplayString()
    d BACK^%ETN
}
q
```

Otestujeme, ale @#%\$%^&! V logu nič nie je! Práve sme zistili, že mechanizmus **THROW** nie je kompatibilný s pôvodným mechanizmom zachytávania výnimiek **ZTRAP**. Ako z toho von? Našťastie je pomoc vcelku jednoduchá, aj keď trochu nesystémová. Upravíme ľahko predchádzajúci kód:

```
try {
    s tOsoba##class(tt.osoba).%OpenId(100,4,.tSC)
    //if
    $$System.Status.IsError(tSC) throw
    ##class(%Exception.StatusException).CreateFromStatus(tSC)
    if $$System.Status.IsError(tSC) s ex##class(%Exception.StatusException).CreateFromStatus(tSC) ztrap "ZSC"
} catch (ex) {
    w ex.DisplayString()
    d BACK^%ETN
}
q
```

Teraz sa už všetko správa tak, ako má. Máme objekt reprezentujúci výnimku aj zápis do systémového logu. Navyše môžeme v try – catch sekcii vyrábať rôzne typy chýb pomocou **ztrap** "mojkodchyby" a v sekcii **catch** tieto chyby podľa kódu filtrovať a ošetrovať rôznym spôsobom.

V nasledujúcej časti sa budeme venovať testovaniu kódu pomocou **Unit Testing** nástroja Caché. Ďalšie zaujímavé časti seriálu **Tipy a triky s Caché**, ktoré vychádzajú už od roku 2005, nájdete na stiahnutie tu: <http://www.intersystems.cz/education/university/serialy/typyTrikyCache.html>



DAN KUTÁČ, InterSystems

Oracle bude na vývoji Javy pracovať spoločne s IBM

Oracle a IBM, inokedy dvaja veľkí rivali, sa dohodli na vzájomnej spolupráci pri vývoji **OpenJDK**, hlavnej referenčnej implementácie Javy. Spolupráca sa má primárne týkať **OpenJDK** (Java Development Kit), ale firmy sa chcú spoločne zaoberať aj jazykom Java či rozhraním **Java Runtime Environment**. Obe spoločnosti zároveň zdôraznili, že konzorcium **JCP** (Java Community Process) má hrať pri vývoji Javy aj naďalej hlavnú rolu, jeho význam chcú dokonca ešte posilniť. Podľa čelných predstaviteľov Oraclu je tento krok jasnou ukážkou toho, akým smerom sa bude vývoj Javy ďalej uberať. Po prevzatí spoločnosti **Sun Microsystems** Oraclom sa totiž veľa používateľov Javy strachovalo o jej osud. Postupne sa však ukazuje, že išlo o zbytočné obavy. Java tu stále je a Oracle s ňou má očividne veľké plány.

„Naším cieľom je zlepšiť prínos **OpenJDK** pre celú komunitu Javy a zrýchliť tak aj tempo inovácií,“ hovorí **Rod Smith**, viceprezident Oraclu pre nové technológie. Spoločnosť **IBM** sa na vývoji **OpenJDK** v minulosti nikdy nezúčastnila, pracovala však na alternatívnej open source implementácii Javy, projekte **Apache Harmony**. Pri tom plánuje **IBM** zostať aj naďalej, hlavné úsilie však podľa Smitha bude smerovať do vývoja **OpenJDK**. Oracle počíta s dodržaním nedávno oznámeného vývojového plánu pre **JDK 7** a **8**. Siedma verzia má priniesť len zlepšenia niektorých funkcií a takisto nové rozhrania **I/O**, zatiaľ čo **JDK 8** vnesie aj menšie zmeny do jazyka a skráti čas potrebný na spustenie **JVM**. Analytik **AI Hilwa** z **IDC** vníma spojenectvo „dvoch najväčších Java hráčov“ ako veľmi pozitívne. Podľa neho je tak zaručené, že vývoj Javy bude pokračovať, navyše oveľa rýchlejšie. „Oracle sa pre vývoj Javy veľmi nadchol, vyzerať to, že je preňho veľmi dôležitá. To je dobrá správa hlavne pre vývojárov v jazyku Java,“ hovorí Hilwa.