

A Java JSP, JavaBeans és RMI eszközei. CORBA és alkalmazása Java környezetben

Java Server Pages (JSP)

- (Az Active Server Pages ASP és a PHP-hez hasonló technológia)...
- Statikus és dinamikus részek közös állományban (.jsp)
- Statikus→HTML/XML... Dinamikus→ugyanitt, a statikus részből speciális tagok hivatkoznak rájuk
- Jsp oldalak→servlet lesz belőlük
- Előre definiált Java objektumkörnyezet...

Java Remote Method Invocation (RMI)

- (Remote Procedure Call (RPC) – távoli eljáráshívás)
- Remote Method Invocation (RMI) – távoli módszerhívás – objektumorientált világban
- Szállítási réteg:
 - Bájtsorozatok szállítása címterületek/gépek között
 - Távoli objektumok táblázata (gép, cím stb.)
 - Alapvetően TCP alapú szállítás
- Távoli referencia réteg:
 - Protokollok illesztése (különböző kapcsolatfelvételi és válaszolási stratégia), hivatkozás perzisztens objektumokra
- Váz réteg:
 - Ügyfél oldali csont (a kiszolgáló kód elolvasása alapján proxy generálás)
 - Kiszolgáló oldali váz
 - Eljuttatja a hívást és a paramétereket az alsóbb rétegekhez
- Osztott objektumorientált alkalmazás
 - Hogyan találjuk meg a távoli objektumokat?
 - Hogyan kommunikálunk velük?
 - Hogyan tölthetjük be a használt objektumok definícióját?
- 1. Az RMI kiszolgáló meghívja a saját gépén futó rmiregistry programot és regisztráltatja magát
- 2. Az RMI ügyfél a hálózaton keresztül meghívja a rmiregistry regisztrációs adatbázist, aki átadja a távoli kiszolgáló objektumot
- 3. Az RMI ügyfél meghívja a megkapott RMI kiszolgáló eljárásait
- Ügyfél oldal – objektum csont (stub), amely rendelkezik a valóságos objektum minden nyilvános módszerével.
 - Becsomagolja a hívást és a paramétereket (marshalling)
 - kapcsolatba lép a távoli Java géppel, és meghívja a kiszolgáló módszereket, megvárja(!) amíg a művelet befejeződik
 - a megérkezett visszatérési értékeket kicsomagolja (unmarshalling), és visszaadja a hívónak
- Kiszolgáló oldal – kiszolgáló (csont)váz (skeleton)
 - fogadja a kérést, kicsomagolja, a paramétereket (unmarshalling)
 - meghívja az eljárásmegvalósítást
 - az eredményeket kicsomagolja (marshalling) és visszaküldi a hívónak.
- Osztott szemétygyűjtés: a Java referenciaszámlálás szemétygyűjtési megoldása különböző gépek/címterek között
- A java.rmi.Remote interface-t kibővítjük, majd implementáljuk... A lefordított .class állományból az rmic program segítségével hozunk létre csontot és csontvázat
- Paraméterátadás, Naming
 - Java objektumok sorosítása (serialization)

- Ügyfél és kiszolgáló egymásra találása: java.rmi.Naming/„tudakozó” szolgáltatás.
- Regisztrációs adatbázis
 - rmiregistry elindít az adott gépen egy regisztrációs szolgáltatást

Az RMI programozása

- Mi a hasonlóság a helyi és távoli objektumok között?
 - Hívásparaméterként objektumok (-referenciák is átadhatók)
 - Objektumok és felületek (interface) egymásba konvertálhatók
 - Az instanceof operátor és a típuskonverzió (casting) tökéletesen működik (meghatározhatók vele a megvalósított felületek)
- Mi a különbség a helyi és a távoli objektumok között?
 - Ügyféloldaltól a kiszolgáló felületei szólíthatók meg, nem az implementáló osztályai
 - Távoli hívások argumentumai másolással adódnak át
 - Távoli objektumok argumentumai hivatkozás szerint adódnak át
 - Az Object osztály néhány eljárása távoli objektumokra változik
 - Ügyféloldalon a hálózati hibákat is kezelni kell

Common Object Request Broker Architecture (CORBA)

- OMG – Object Management Group (www.omg.org)
- OMA – Object Management Architecture
 - Common Object Request Broker Architecture, egy implementációja egy ORB...
- Objektum-orientált távoli hívások kezelése (middleware)
- Előnyök:
 - Nyílt szabvány
 - Objektum orientált
 - Helyszín átlátszóság (többféleképpen telepíthető)
 - Platform- nyelv-, megvalósításfüggetlenség (Ada, Cobol, C, C++, Smalltalk, Lisp, Prolog, Java)
 - Kapcsolat más middleware rendszerekkel (RPC, (D)COM, RMI)
 - Szinkron és aszinkron hívások

CORBA:

A CORBA elemei

- Interface Definition Language (IDL)
 - C++ (Java)-hoz hasonló nyelv
 - Platformfüggetlen
 - Fordító: *.idl → csonk, váz kód (stub, skeleton)
- Adatátviteli protokollok:
 - General Inter-ORB Protocol (GIOP): alapprotokoll, amely az adatok/bitek/bájtok szintjén írja le a kommunikációt
 - Internet Inter-ORB Protocol (IIOP): GIOP-ra alapuló protokoll Internetes üzenetek cseréjére (TCP-IP alapon)
 - További protokoll részletek nyitottak

Tulajdonságai:

- Nyelvfüggetlenség
- Távoli objektumhivatkozások:
- Mutató-szerű szerkezetek, amelyek másik folyamatba (másik gépre) mutatnak
 - Objektumpéldányra vonatkoznak
 - Különböző hivatkozások jelezhetik ugyanazt az objektumot
 - Üres és érvénytelen hivatkozások
 - Erősen típusos hivatkozások

- Késői típuslekötés (alaptípusra hivatkozás futásidőben hivatkozhat leszármaztatott típusra is)
- Hivatkozások persistent (háttérben tárolt) v. tranziens (ideiglenes) objektumokra
- Hivatkozások szabványosak (Interoperable Object Reference, IOR)

CORBA \leftrightarrow RMI

Corba:

- Platformfüggetlen technológia
- IDL felületleírás
- Több megvalósítás is létezik
- Szinkron/aszinkron hívások
- Igen széles szolgáltatási környezet
- Nincs késői típuslekötés

RMI:

- Csak Java technológia
- Java felületleírás
- Csak a Sun megvalósítást használjuk
- Csak szinkron hívások (megvárja a lefutást)
- Java környezet
- Minden Java lehetőség elérhető (típuslekötés is)