



Funkciógazdag,
webszolgáltatást használó Java ügyfél
összeállítása

Tartalom

Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása . . . 1

Bevezetés: Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása	1
1. modul: Az ügyfél grafikus felületének tervezése a vizuális szerkesztőben	3
1.1 feladat: Java projekt beállítása	3
1.2 feladat: Alkalmazottak tábla hozzáadása és elrendezése	4
1.3 feladat: Vizuális osztály futtatása	8
2. modul: Vizuális összetevők webszolgáltatáshoz kötése	10
2.1 feladat: Webszolgáltatás telepítése és elhelyezése	10
2.2 feladat: Az alkalmazottak tábla a webszolgáltatás adatforrásához kötése	12

2.3 feladat: Részletmezők kötése a táblakijelöléshez	17
2.4 feladat: A Frissítés gomb egy műveletgyűjtőhöz kötése	21
2.5 feladat: A Törlés gomb és a megerősítés párbeszédablak engedélyezése	23
2.6 feladat: Műveletek és kötések beállítása új alkalmazott felvételéhez	25
2.7 feladat: A Mégse gomb viselkedésének programozása	30
2.8 feladat: Szűrő beállítása az alkalmazottak táblára	31
Összegzés: Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása	32

Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása

Ez az ismertető megtanítja a vizuális Java szerkesztő használatára, amivel funkciógazdag, webszolgáltatást használó Java ügyfelet állíthat össze. Az ismertetőben felépített ügyfél neve Cégcímár.

A Cégcímár a cég alkalmazottainak címárát fenntartó Java alkalmazás. Az alkalmazás az alkalmazottak rekordjainak létrehozását, lekérdezését, frissítését és törlését intéző metódusokat biztosító minta-webszolgáltatáshoz csatlakozik.

Az ügyfél vizuális összeépítése a vizuális Java szerkesztőben történik, Swing összetevők használatával. A vizuális Java szerkesztő segédosztály-készleteket (adatforrások, adatobjektumok és összekötők) biztosít, vagy összeköttetést a webszolgáltatás kezeléséhez. A webszolgáltatás az IBM WebSphere Application Server 6.0 telepítésével együtt van helyben telepítve, és az eszközök segítségével létrehozhat egy Webszolgáltatás leírónyelv (WSDL) fájlra alapuló Java proxy-t.

Lásd a befejezett terméket

Feladat célja

Ebben az ismertetőben a következő részek szerepelnek:

- A Vizuális Java szerkesztő használata felhasználói felület tervezéséhez és elrendezéséhez
- Felületi elemek összekötése adatobjektumokkal és webszolgáltatással

Szükséges idő

2 óra 15 perc

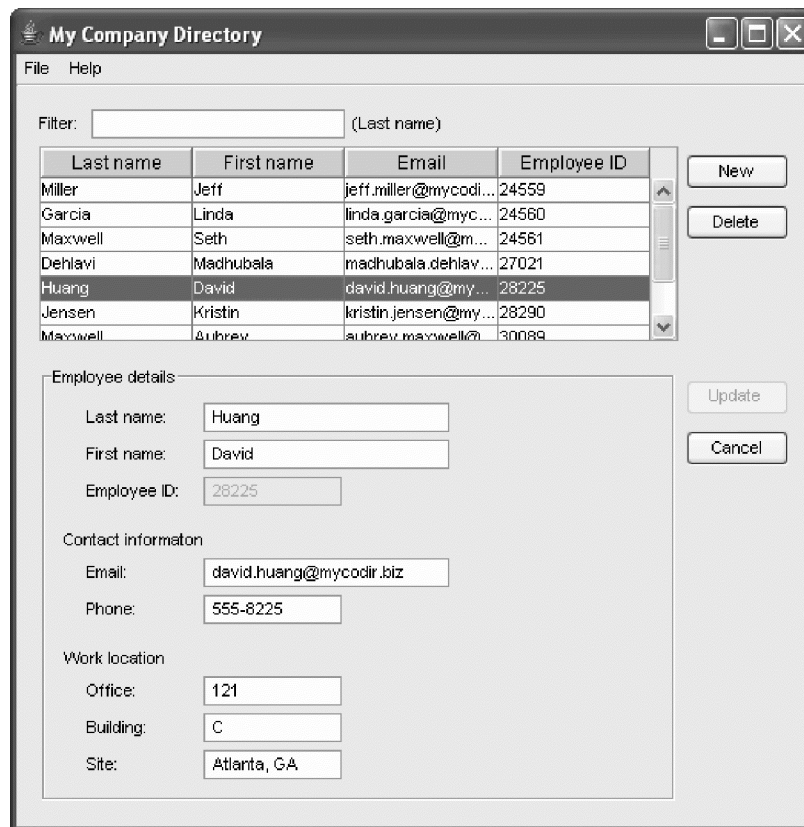
Kapcsolódó tájékoztatás

PDF változat megtekintése

Ismertető: Hello World Java

Bevezetés: Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása

Az ügyfél grafikus felhasználói felülete (GUI) jórészt előre fel van építve a Swing összetevőkből. Az első modulban a vizuális Java szerkesztővel be fogja fejezni a grafikus felület lényeges elemeinek elhelyezését. A második modulban a grafikus felület összetevőit fogja a webszolgáltatás adatforrásához, szolgáltatásaihoz és az adatforrás által visszaadott objektumokhoz kötni. Az alkalmazásnak a vizuális Java szerkesztőben felépítésekor olyan adatforrásokat, adatobjektumokat és kötéseket fog használni, melyek a vizuális Java szerkesztő által létrehozott segédosztályok egyedei és melyeket az alkalmazás használ.



A kész termék képe:

Feladat célja

Ebben az ismertetőben a következő részek szerepelnek:

- A Vizuális Java szerkesztő használata felhasználói felület tervezéséhez és elrendezéséhez
- Felületi elemek összekötése adatobjektumokkal és webszolgáltatással

Szükséges idő

Az ismertető befejezéséhez megközelítőleg két és fél óra szükséges.

Rendszerkövetelmények

- WebSphere Application Server 6.1. A kiszolgálón már telepítve lehet a termék, vagy használhat saját önálló telepítést is. Az ismertető példahelyzetében telepítenie kell egy minta webszolgáltatást a saját gépen futó WebSphere alkalmazáskiszolgálóra.

A minta webszolgáltatás futhat más kiszolgálókon is, de az ismertető csak a WebSphere Application Server 6.0 és 6.1 változatával került tesztelésre.

Előfeltételek

Ismernie kell a következő fogalmakat:

- Alapvető Java fejlesztés
- Alapvető webszolgáltatási elvek
- Alapszintű munkaterületkezelés, például projektek kezelése, illetve navigációs perspektívák és nézetek

1. modul: Az ügyfél grafikus felületének tervezése a vizuális szerkesztőben

Ez a modul bemutatja, hogy hogyan lehet a vizuális Java szerkesztővel vizuális összetevőt adni egy alkalmazáshoz, illetve hogyan lehet az összetevőt vizuálisan elrendezni és az elrendezés megszorításait beállítani. A modul utolsó feladatként megmutatja a Java fájl futtatását, hogy megtekinthesse a végeleges alkalmazás tényleges megjelenését.

Ne feledje: A modul elkezdése előtt rendelkeznie kell az ismertető bevezetésében körvonalazott tudással.

Feladat célja

A modul feladatainak elvégzése után meg fogja érteni a fogalmakat, illetve el tudja végezni az alábbiakat:

- JTable hozzáadása és elrendezése Java felületen
- Vizuális osztály futtatása a munka tesztelésére

Szükséges idő

A modul elvégzése megközelítőleg 15 percet vesz igénybe.

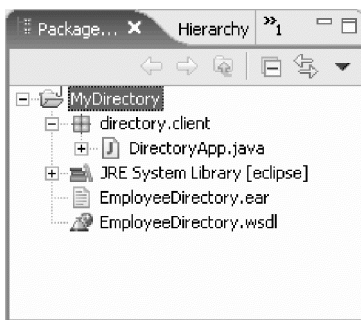
1.1 feladat: Java projekt beállítása

Ebben a leckében a Címtár projektet fogja beállítani egy projekt importálásával a munkaterületre. A projekt egyetlen Java osztályt, valamint egyéb később használandó fájlokat tartalmaz.

Mivel ennek az ismertetőnek lényege a vizuális komponensek kötése egy webszolgáltatáshoz, így a "Saját vállalati címtár" alkalmazás legtöbb Java grafikus felhasználói felülete már előre meg van tervezve.

A Címtár projekt az ismertető során elsősorban használt Java projekt. Tartalmazza a DirectoryApp.java fájlt, ami a felépítendő fő Java alkalmazást tartalmazó Java fájl. Az ismertető számos a Címtár projekt számos verzióját tartalmazza. Minden egyes modul elindításához tartozik egy verzió, és ezenkívül megtalálja a befejezett projekt kész verzióját is.

1. Importálja a MyDirectory projektet.
2. Győződjön meg róla, hogy a Java perspektíva Csomagböngészőjében a MyDirectory projekt az alábbi képhez hasonlóan néz ki:



Előadás ellenőrzési pont

Ebben a leckében a Címtár példaprojektet importálta, amely az ismertető kezdőpontjaként használható.

A Címtár projekt az alábbi erőforrásokat tartalmazza:

- DirectoryApp.java: Az ismertető során fejlesztett alkalmazást tartalmazó Java fájl. A DirectoryApp.java fájl a directory.client nevű Java csomagban van.
- EmployeeDirectory.ear: A minta-webszolgáltatást tartalmazó vállalati alkalmazás. A 2. modulban ezt a webszolgáltatást fogja a helyi WebSphere Application Server 6.0 változatra telepíteni.

- EmployeeDirectory.wsdl: Egy XML fájl, ami a Webszolgáltatás leíró nyelv (WSDL) használatával írja le a telepítendő minta-webszolgáltatást. A 2. modulban ezzel a WSDL fájjal fog Java proxy-t előállítani az alkalmazáshoz.

1.2 feladat: Alkalmazottak tábla hozzáadása és elrendezése

Ebben a feladatban egy JScrollPane és egy JTable osztályt ad az alkalmazáshoz a vizuális Java szerkesztővel. A későbbi feladatokban be fogja programozni a JTable osztályt, hogy olyan webszolgáltatásból vegye az adatokat, ami a céges címtár összes alkalmazottjának listáját adja vissza.

A JTable felvétele után a vizuális Java szerkesztő tervezés nézetével személyre szabja a JTable elrendezését, az alábbi meghatározások szerint:

- A JTable vízszintesen három, függőlegesen két cellára terjedjen ki
- Legyen 15 pixeles behúzás a bal oldalon
- A JTable át lesz nevezve employeesTable névre.

Megjelenés

DirectoryApp.java fájl megnyitása a vizuális Java szerkesztőben

A DirectoryApp.java fájl a vizuális Java szerkesztőben megnyitásához tegye a következőket:

1. A Java perspektíva Csomagböngésző nézetében bontsa ki a MyDirectory projektet és a directory.client csomagot.
2. Kattintson a jobb egérgombbal a DirectoryApp.java fájlra, majd válassza az előugró menü **Társítás** → **Vizuális szerkesztő** menüpontját. A vizuális Java szerkesztő betölti a Java osztályt, és megjeleníti a látványtervet a grafikus rajzolási területen.

Tipp:

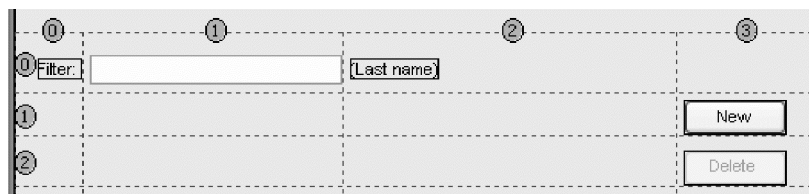
- A vizuális Java szerkesztő kinézetének megváltoztatásához válassza az **Ablak** → **Beállítások** → **Java** → **Vizuális szerkesztő** menüpont Swing kinézet lehetőségét. A beállítás az osztály következő megnyitásakor lép életbe. Ez az ismertető a Windows kinézetet használja.
- A Vizuális szerkesztő alapértelmezett Java fájl szerkesztővé beállításához kattintson az **Ablak** → **Beállítások** menüpontra, és a **Munkaterület** → **Fájlok társítása** oldalon adja meg a kívánt beállítást.

JTable hozzáadása JScrollPane összetevőhöz

A DirectoryApp.java főablaka JFrame összetevőt használ, fő tartalom panelként a JPanel használatával. Ebben az alkalmazásban a JPanel neve JPanel. A JPanel a GridBagLayout nevű elrendezéstípust használja. A GridBagLayout hatékony elrendezési séma, ami vizuális összetevőkkel benépesíthető cellarácsokon alapszik. A vizuális Java szerkesztő megjeleníti a rácsok keretét, így megkönnyíti a GridBagLayout kezelését. Elhelyezési jelzőket is megjelenít, amikor új összetevőket helyez a rácsra, illetve kezelőket a GridBagLayout összetevőinek mozgatasakor vagy méretezésekor.

Az alkalmazottak (egy javax.swing.JTable) tábla a DirectoryApp.java felhasználói felülethez adásához tegye a következőket:

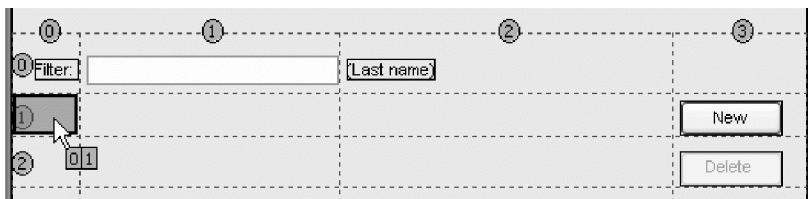
1. Kattintson a jobb egérgombbal a JPanel összetevőre a tervezés nézetben vagy egy Java komponens nézetben, majd válassza az előugró menü **Rácsok megjelenítése** menüpontját. Vörös pontozott vonal jelzi a rács szegélyét, számokkal ellátott körök pedig a sorok és oszlopok számát. Például az **Új** gomb az 1. sorban (y rács) és a 3. oszlopban (x rács) van.



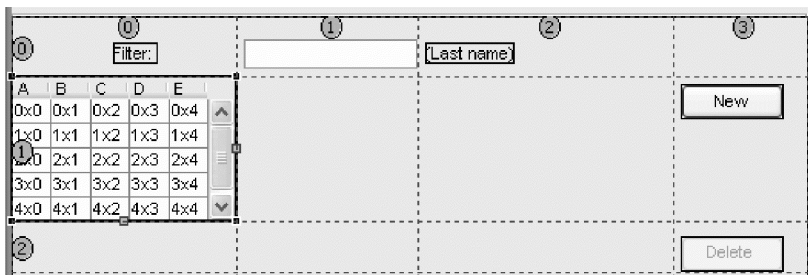
2. A vizuális Java szerkesztő palettán válassza a **JTable JScrollPane összetevő** Swing összetevőt, ami a paletta **Swing összetevők** fiókjában van kategorizálva.

Tipp: A paletta alapértelmezésben a tervezési terület jobb oldalán van összehúzza. Átméretezheti és mozgathatja a palettát.

3. Helyezze az egérmutatót a 0. oszlop 1. sorában levő cella fölé:



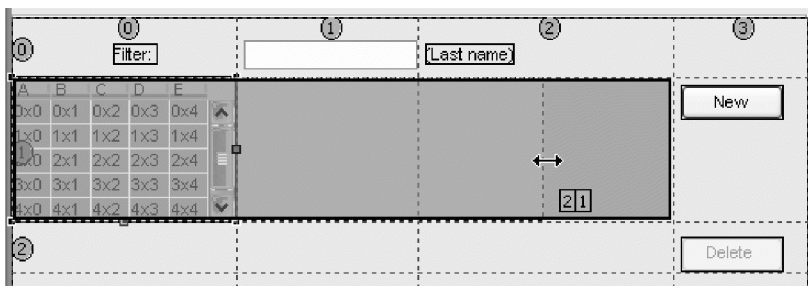
- Ahogy a rács fölé mozgatja az egérmutatót, két számozott négyzet jelenik meg, bennük az adott rács x és y koordinátájával.
 - Ha közvetlenül a rács szegélye fölé viszi az egérmutatót, akkor új sorokat és oszlopokat hozhat létre, a meglévő sorok és oszlopok pedig újraszámozódnak. Ilyen esetben sárga négyzetek jelennek meg az egérmutatón, továbbá sárga sávok a rácsok közt, illetve sárga oszlop- és sorcímkek jelzik ezt a viselkedést, valamint az új elhelyezés hatását.
4. Kattintson a bal egérgombbal, hogy lerakja a JScrollPane és a JTable összetevőt a 0. oszlop 1. sorába:



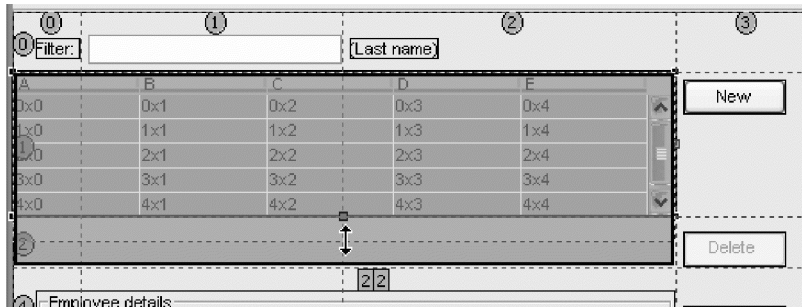
a JScrollPane és JTable összetevők kiterjesztése a rács több oszlopára és sorára

Terjessze ki a JScrollPane összetevőt (és leszármazottját, a JTable összetevőt) három oszlopra és két sorra a jobb térköz és méretezési viselkedés érdekében. A tábla kiterjesztéséhez tegye a következőket:

1. Jelölje ki a JScrollPane összetevőt a tervezési területen vagy a Java komponens nézetben (mivel épp most adta hozzá, ki van jelölve). Figyelje meg a JScrollPane jobb oldalán és alján látható zöld négyzeteket. Ezekkel az újraméretezési kezelőkkel fogja áthúzni a JScrollPane összetevőt több oszlopra és sorra.
2. Kattintson a bal egérgombbal a JScrollPane jobb oldalán található zöld kezelőre, és tartsa lenyomva a gombot.
3. Húzza jobbra az egérmutatót, míg az elhelyezés átkerül a 2. oszlop 1. sorába. Sötétszürke terület jelzi azokat a cellákat, amiket az összetevő elfoglal az egérgomb felengedése után.



4. Engedje el az egérgombot. A JScrollPane most három oszlopra terjed ki.
5. Ismételje meg ehhez hasonlóan a folyamatot: húzza a JScrollPane alján levő kezelőt, míg ki nem terjeszti a 2. sorra:



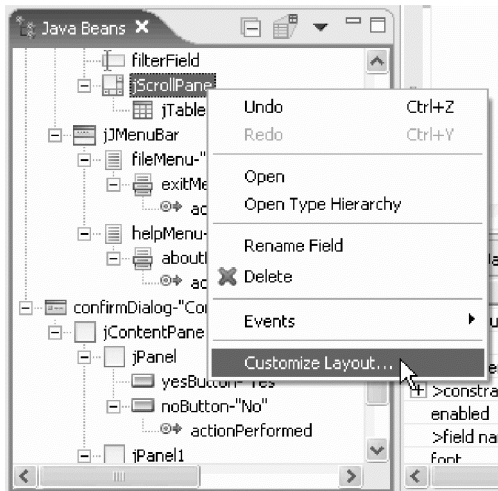
A JScrollPane térközeinek személyre szabása a GridBag rácsban

A GridBagLayout kezelő egy másik jellemzője az, hogy az elrendezés további személyre szabásához különböző megszorításokat adhat meg. Megadhatja például a következő megszorításokat:

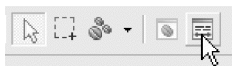
- **horgony:** Az összetevők kaphatnak a cellán belül horgony tájolást, ami akkor adja meg az összetevő mozgását, amikor a felhasználó átméretezi az alkalmazást. Lehorgonyozhat például egy összetevőt a bal felső, a balközép, a jobb alsó sarokba vagy középre.
- **kitöltés:** Megadható, hogy egy összetevő kitöltsön minden szabad helyet a cellában vízszintesen vagy függőlegesen.
- **behúzások:** Megadható az összetevő felső, alsó bal és jobb oldali kitöltése, hogy térköz legyen az összetevő és a rács szegélye közt.

A JScrollPane horgonyzásának, kitöltésének és behúzásának személyre szabásához tegye a következőket:

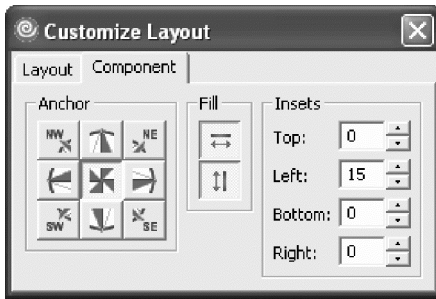
1. Kattintson a jobb egérgombbal a JScrollPane összetevőre a tervezés nézetben vagy a Java komponens nézetben, majd válassza az előugró menü **Elrendezés személyre szabása** menüpontját.



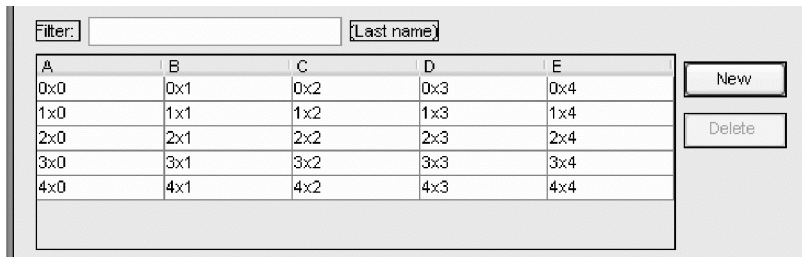
Tipp: Az Elrendezés személyre szabása párbeszédablak nyitva maradhat, míg kiválasztja és módosítja a különböző összetevők elrendezését. Bármikor megnyithatja az Elrendezés személyre szabása párbeszédablakot a menüsor megfelelő gombjára kattintva:



2. Ellenőrizze, hogy az Elrendezés személyre szabása párbeszédablak Összetevők lapján a Horgony középre gomb van lenyomva.
3. Győződjön meg róla, hogy mind a **Vízszintes kitöltés**, mind a **Függőleges kitöltés** gomb le van nyomva.
4. Vegyen fel 15 (pixel) bal oldali behúzást, hogy a JScrollPane bal oldali térköze hasonló legyen az alkalmazás többi vizuális összetevőjéhez.



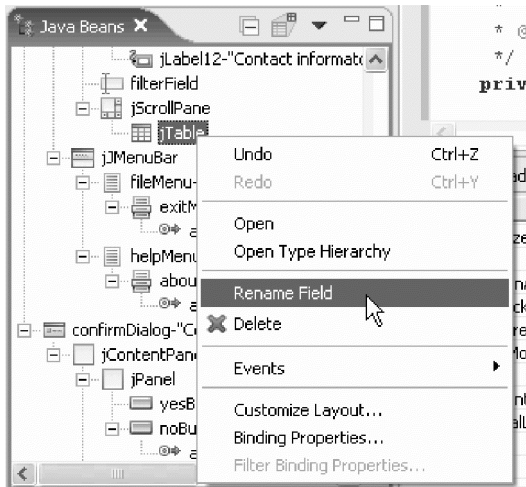
A tábla például már egy vonalban van a **Szűrő** címkével.



Az új JTable összetevő átnevezése és beállítása egy sor kiválasztására

Mivel a későbbiekben használni fogja a táblát, hasznos átnevezni a JTable példányt és annak lekérdező metódusát. A tábla átnevezése:

1. Kattintson a jobb egérgombbal a JTable összetevőre a Java komponens nézetben, majd válassza az előugró menü **Mező átnevezése** menüpontját.



2. Írja be az `employeesTable` értéket és kattintson az **OK** gombra. A JTable neve már `employeesTable`, a példányosító metódusé pedig `getEmployeesTable`.
3. Állítsa be, hogy a tábla csak egy sort jelölhessen ki:
 - a. Jelölje ki az `employeesTable` táblát a tervezés nézetben.
 - b. Válassza ki a Tulajdonságok nézetben a **selectionMode** tulajdonságot, és állítsa `SINGLE_SELECTION` értékre.

Property	Value
preferredSize	375,80
rowHeight	16
rowSelectionAllowed	true
selectionBackground	49,106,197
selectionForeground	Color:white
selectionMode	SINGLE_SELECTION
showGrid	
showHorizontalLines	true
showVerticalLines	true
toolTipText	
visible	true

c. Mentse el a DirectoryApp.java fájlt.

Előadás ellenőrzési pont

Ebben a leckében megtanulta, hogyan lehet a vizuális szerkesztővel táblát felvenni meglévő felhasználói felülethez. Ezután megtanulta személyre szabni annak elrendezését, helyzetét és térközeit.

1.3 feladat: Vizuális osztály futtatása

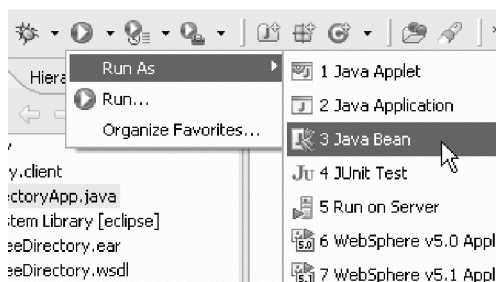
Készen áll a Java alkalmazás futtatására, hogy megnézze a megjelenését. A munkaterület és a vizuális szerkesztő megkönnyíti az alkalmazás gyors futtatását, és ezeket a lépéseket megismételheti a fejlesztés bármely szakaszában, hogy tesztelje az osztály tényleges futási megjelenését és viselkedését.

Megjelenés

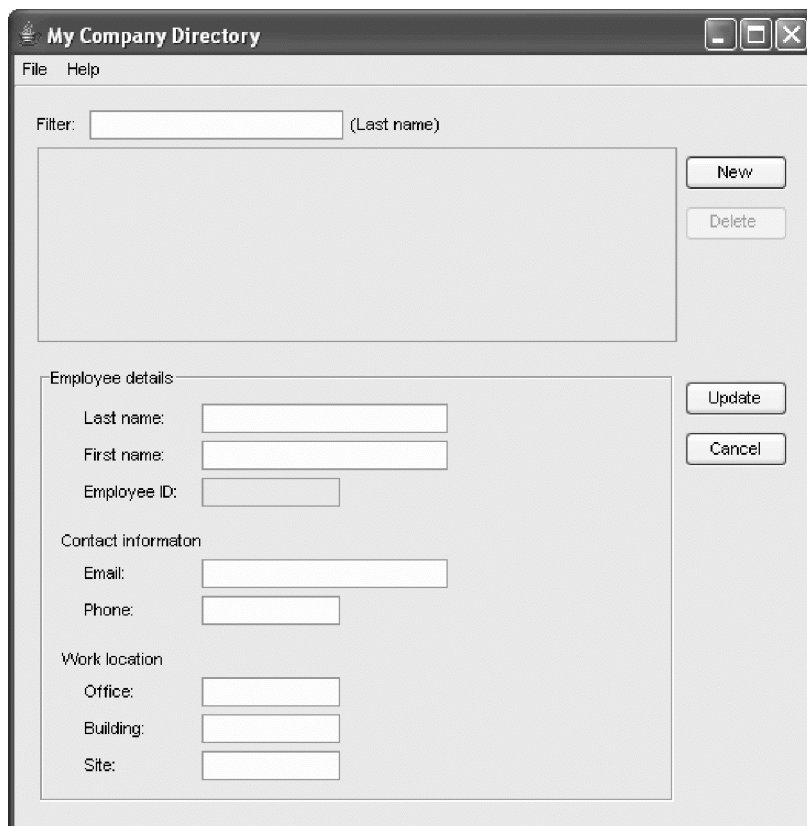
A vizuális Java szerkesztő biztosít egy Java komponens indítót, ami képes main() metódus nélkül futtani az osztályokat. A vizuális osztály futtatásakor egy önálló virtuális gépen (VM) indítja az alkalmazást. Ha Java alkalmazásként futtatja a vizuális osztályt, akkor az indító megkísérli végrehajtani az osztály main() metódusát. Ebben az ismertetőben az alkalmazás tartalmaz egy main() metódust, ami meghívja és megjeleníti a DirectoryApp JFrame összetevőjét, így futtathatja alkalmazásként és Java komponensként is.

A DirectoryApp.java fájl Java komponensként futtatása:

1. Győződjön meg róla, hogy a DirectoryApp.java fájl meg van nyitva a vizuális Java szerkesztőben.
2. Kattintson a menüsor **Futtatás** → **Futtatás másként** → **Java komponens** menüpontjára.



Tipp: Az alkalmazás a munkaasztalon nyílik meg, a Vizuális szerkesztő beállításokban megadott Swing megjelenés használatával (Ablak → Beállítások → Java → Vizuális szerkesztő). Ennek alternatívájaként kattintson a **Futtatás** → **Futtatás másként** menüpontra, és adja meg az indítási konfigurációt ennek a Java komponensnek az indításához. Ha az alkalmazást komponens helyett alkalmazásként futtatja, akkor a Windows megjelenést is használni fogja, mivel az van megadva a main() metódusban. Az ismertető képei a Windows megjelenést mutatják.



Előadás ellenőrzési pont

Mivel a felületnek még csak a terve van kész, de nincsenek adatkapcsolatok vagy események programozva, nem tehet semmit az alkalmazással. Azonban megtekintheti az alapvető elrendezést és megjelenést, ahogy azt a felhasználó is látni fogja. Rákattinthat a gombokra, de észre fogja venni, hogy semmi nem történik. A Fájl és Súgó menü azonban már meg van valósítva. Kirpóbálhatja, hogy mit tesznek, és megvizsgálhatja a Java kódot, hogy megfigyelje, hogyan történt a megvalósításuk az actionPerformed eseményekkel.

Tanultak

Ez a modul megismertette a funkciógazdag ügyfél felületének vizuális Java szerkesztővel tervezésével. Az ügyfél vizuális megjelenésének tervezésén túl azonban jóval többet kell tenni az ügyfél hasznossá tételéhez. Általában az események kezelését és egyéb funkciókat kell megadnia, illetve ebben az esetben a vizuális elemek kötését az adatforráshoz.

Ebben a modulban az alábbiak elvégzését tanulta meg:

- Java projekt importálása Projekt adatsere importálással
- JTable hozzáadása vizuális osztályhoz a JScrollPane összetevőn
- A GridBagLayout kezelő használata a funkciógazdag ügyfél táblájának vizuális elrendezéséhez
- Az alkalmazás futtatása a Java ügyfél tényleges megjelenésének megtekintéséhez

A következő modulban (2. modul: Vizuális összetevők webszolgáltatáshoz kötése) hatékony ügyféllé változtatja a Cégcímtár ügyfelet, ami webszolgáltatás metódusokkal hoz létre, kérdez le, frissít és töröl alkalmazott rekordokat a céges címtárból.

2. modul: Vizuális összetevők webszolgáltatáshoz kötése

A modul megtanítja a Cégcímtár vizuális elemeinek (gombok, az alkalmazott tábla, mezők és egyéb műveletek) webszolgáltatáshoz kötésére. A webszolgáltatás valós működést biztosít az alkalmazottak mintacímtárban létrehozására, lekérdezésére, frissítésére és törlésére.

Feladat célja

A modul feladatainak elvégzése után meg fogja érteni a fogalmakat, illetve el tudja végezni az alábbiakat:

- Tábla adat-webszolgáltatás adatforrásához kötése
- Mező objektumokhoz kötése
- Gombok programozása műveletekre

A modul befejezéséhez körülbelül **2 órára** van szükség.

2.1 feladat: Webszolgáltatás telepítése és elhelyezése

Ebben a feladatban egy minta vállalati alkalmazást (EAR) fog telepíteni a WebSphere Application Server 6.1 alkalmazáskiszolgálóra, és telepíti az EmployeeDirectory webszolgáltatást. Az alkalmazás ezt a webszolgáltatást az alkalmazottak rekordjainak létrehozására, olvasására, frissítésére és törlésére.

Mielőtt elkezd, el kell végeznie *az alábbi lehetőségek egyikét*, hogy megbizonyosodjon arról, hogy a MyDirectory projekt a megfelelő kezdési stádiumban van:

- Végezze el a “1. modul: Az ügyfél grafikus felületének tervezése a vizuális szerkesztőben” oldalszám: 3 részt.
vagy
- Importálja a MyDirectory projektet a 2. modul kezdőpontján.

Tipp: Ha az importálás során nem ad meg másik projektnevet, akkor felülírja a MyDirectory projekt tartalmát.

A MyDirectory Java projekt tartalmazza az EmployeeDirectory.ear fájlt. A WebSphere adminisztrátori konzollal fogja telepíteni az EmployeeDirectory vállalati alkalmazást, amit az EAR fájl tartalmaz. Az alkalmazás telepítésekor egyben telepíti a vele együtt érkező webszolgáltatást is. A befejezett Cégcímtár alkalmazás ezt a telepített webszolgáltatást használja.

Az EmployeeDirectory mintaalkalmazás telepítéséhez és a webszolgáltatás a WebSphere Application Server 6.1 környezetre telepítéséhez tegye a következőket:

1. Indítsa el az alkalmazáskiszolgáló egy példányát a munkaterületről. Többféleképpen is elindíthatja a kiszolgálót, de az alábbi lépések az munkaterületről indítást írják le:
 - a. Nyissa meg a Kiszolgálók nézetet. A Kiszolgálók nézet a Java perspektívához adásához kattintson az **Ablak → Nézet megjelenítése → Egyéb és kiszolgáló kijelölése → Kiszolgálók** menüpontra.
 - b. A Kiszolgálók nézet a telepített és beállított kiszolgálók listáját jeleníti meg.
 - c. Kattintson a jobb egérgombbal a kiszolgálóra, majd válassza az előugró menü **Indítás** menüpontját. Amikor a Kiszolgálók nézetben a kiszolgáló állapota **Elindult**, vagy amikor a konzolon a **Server1** kiszolgáló megnyitva **e-business**hez állapot jelenik, akkor a kiszolgáló sikeresen elindult. Most már futtathatja az Adminisztrátori konzolt.

Megjegyzés: Ha nincs kiszolgálópéldány a Kiszolgálók nézetben, akkor hozzon létre új kiszolgálót:

- a. Kattintson a jobb egérgombbal a Kiszolgálók nézetben, majd válassza az előugró menü **Új → Kiszolgáló** menüpontját.
 - b. A WebSphere Application Server 6.1 hozzáadásához használja az **Új kiszolgáló** varázslót.
2. Futtassa a WebSphere adminisztrátori konzolt. Az Adminisztrátori konzol futtatásának is több módja van, de ez az útmutatás a munkaterületről indítást írja le:

- a. A Kiszolgálók nézetben kattintson a jobb egérgombbal az elindított kiszolgálóra, majd válassza az előugró menü **Adminisztrátori konzol futtatása** menüpontját. A WebSphere adminisztrátori konzol egy böngészőablakban nyílik meg.
 - b. Adjon meg egy felhasználói azonosítót és kattintson a **Bejelentkezés** gombra. Megjelenik az Adminisztrátori konzol üdvözlőoldala. A megadott felhasználói azonosító csak arra szolgál, hogy nyomon lehessen követni a felhasználóhoz tartozó módosításokat a kiszolgáló konfigurációs adataiban.
3. Az Adminisztrátori konzollal telepítse a MyDirectory projektben található EmployeeDirectory.ear vállalati alkalmazást. Az Adminisztrátori konzol varázsló megközelítéssel segíti az alkalmazások telepítését, azaz a **Tovább** gombra kattintva lépked egyik oldalról a következőre, míg minden beállítást megad. Az ismertető webszolgáltatását tartalmazó minta vállalati alkalmazás telepítéséhez tegye a következőket:
- a. Az Adminisztrátori konzol bal oldalán bontsa ki az **Alkalmazásmenü** lehetőséget, és kattintson az **Új alkalmazás telepítése** lehetőségre.
 - b. Válassza a **Helyi fájlrendszer** lehetőséget, és az **Útvonal megadása** mezőben adja meg a MyDirectory projektben levő EmployeeDirectory.ear fájl teljes elérési útját. Tipp: A teljes elérési út megismeréséhez kattintson Csomagböngészőben a jobb egérgombbal az EmployeeDirectory.ear fájlra, majd válassza az előugró menü **Tulajdonságok** menüpontját. Az adatlap felsorolja a fájl helyét is, amit kimásolhat és beilleszthet az **Útvonal megadása** mezőbe.
 - c. Kattintson többször a **Tovább** gombra, amíg meg nem jelenik a **Telepítési beállítások kiválasztása** oldal.
 - d. Válassza a **Webszolgáltatások telepítése** lehetőséget.
 - e. Kattintson többször a **Tovább** gombra, míg el nem ér az **Összegzés** oldalra, ahol is kattintson a **Befejezés** gombra.
 - f. Kattintson a **Mentés a mesterkonfigurációba** hivatkozásra, amikor a módosításokat alkalmaznia kell a helyi konfigurációra. Nézze át a módosításokat, és kattintson a **Mentés** gombra.
4. Az Adminisztrátori konzollal indítsa el az EmployeeDirectory alkalmazást:
- a. Kattintson az **Alkalmazások → Vállalati alkalmazások** menüpontra. Az EmployeeDirectory alkalmazás megjelenik a kiszolgálóra telepített alkalmazások felsorolásában, de az állapota Leállított.

Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL		
Refresh Refresh Refresh Refresh		
Select	Name	Status
☐	DefaultApplication	⇒
☒	EmployeeDirectory	✖
☐	Query	⇒
☐	SchedulerCalendars	⇒
☐	filetransfer	⇒
☐	ivtApp	⇒
Total 6		

- b. Válassza ki az EmployeeDirectory melletti jelölőnégyzetet, és kattintson az **Indítás** gombra. Egy üzenet jelzi, hogy az EmployeeDirectory sikeresen elindult, és az Állapot ikon zöld nyílra változik.

Az EmployeeDirectory alkalmazás a saját gép 9080-as portján fut, és elérhetőek a webszolgáltatások. Az ismertető befejezése után visszatérhet ide az Adminisztrátori konzolban, és leállíthatja, majd eltávolíthatja az EmployeeDirectory alkalmazást.

A MyDirectory projektben levő EmployeeDirectory.wsdl fájl megnyitásával (alapértelmezésben nyitva kell lennie a grafikus WSDL szerkesztőben) megvizsgálhatja a most telepített webszolgáltatást. Ha a WSDL fájl nem nyílik meg a WSDL szerkesztőben, akkor lehet, hogy a Webszolgáltatás fejlesztő képesség nincs bekapcsolva a munkaterületen. A munkaterület képességeit a Beállításokban adhatja meg (**Ablak → Beállítások → Munkaterület → Képességek**).

A WSDL szerkesztő alábbi képe az EmployeeDirectory szolgáltatás rendelkezésre álló műveleteit mutatja:



A WSDL szerkesztővel megvizsgálhatja az egyes műveleteket, a vonatkozó kérés üzeneteket és a válaszüzeneteket. Ez segítheti a webszolgáltatás megismerését, és webszolgáltatás a későbbi feladatokban használatának jobb megértését.

2.2 feladat: Az alkalmazottak tábla a webszolgáltatás adatforrásához kötése

A Cégcímtár alkalmazás megjeleníti a címtár aktuális alkalmazotti rekordjainak listáját. A rekordok egy `JTable` (`employeesTable`) összetevőben jelennek meg, oszlopok alapján rendezhetően (vezetéknév, keresztnév e-mail cím és alkalmazotti azonosító alapján). Ahhoz, hogy a rekordok megjelenjenek a táblában, egy olyan adatobjektumhoz kell kötnie az `employeesTable` táblát, amit a minta-webszolgáltatás adatforrása ad vissza.

Megjelenés

Adatobjektumok, adatforrások és kötések áttekintése

Ahhoz, hogy az `employeesTable` tábla kezelni tudjon egy helyi adatobjektumot, a vizuális szerkesztővel adatforrást kell felvenni az alkalmazásba. Az adatforrás a minta-webszolgáltatás proxy-hoz csatlakozik, és feltérképezi az alkalmazás számára elérhető szolgáltatásmetódusokat. Ezután ki kell jelölni a `getLightEmployeeRecord` szolgáltatásmetódust, ami elérhető az adatforrásból. Végül a sor adatobjektumban (`lightEmployeeRecordRows`) visszaadott mezőkhöz köti az alkalmazás `employeesTable` tábláját.

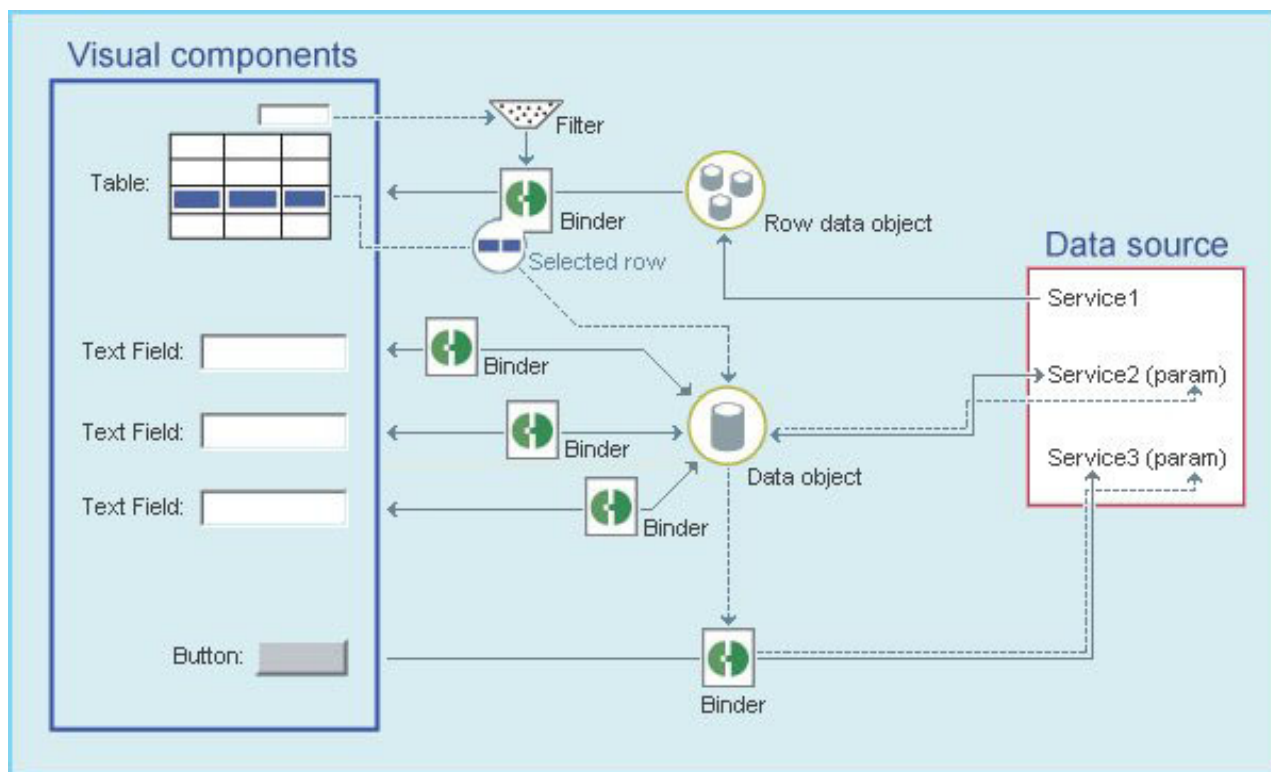
Ezeket az adatforrásokat és adatobjektumokat könnyen és gyorsan létrehozhatja a vizuális Java szerkesztő beépített kötési osztályaival. A vizuális szerkesztő általános felületeket és osztályokat biztosít, melyek létrejönnek a projektben, ahogy a vizuális összetevőket az adatgyárakhoz köti. A kötési osztályok alapértelmezésben a `jve.generated` nevű csomagban lesznek előállítva. A vizuális szerkesztő általános megvalósításként kínálja a kötési osztályokat, amiket személyre szabhat és bővíthet az alkalmazás szükségleteinek megfelelően. Ez az ismertető a már az alapszintű és egyszerű kötési osztályokban is elérhető hatékonyságot és rugalmasságot mutatja be.

Fontos: A feladat megkezdése előtt ajánlott elolvasni a következő sugótémakört. Ezek a témakörök segítik a vizuális Java szerkesztőben biztosított adatobjektumok, adatforrások és kötések mögött húzódó működések és funkciók megismerését.

- Adatkötések bemutatása
- Kötési API leírása

Ebben az ismertetőben webszolgáltatás adatforrást fog használni az alkalmazáshoz, továbbá számos adatobjektumot és kötéstípust. Amikor ezen objektumok példányait felveszi az alkalmazásba, akkor a vizuális szerkesztő hozzáadja a szükséges osztályokat a projekt `jve.generated` csomagjához, amiben bővítheti, kicserélheti vagy újraindítja az adatkötési működést. A vizuális Java szerkesztő vizuális támogatást biztosít az objektumkötésekhez, mivel szabadon alakíthatóan jeleníti meg azt a tervezési területet, ahol az alkalmazásban használt adatobjektumok, adatforrások és kötések láthatóak. A vizuális szerkesztő vonalakat rajzol a vizuális összetevők, illetve az adatobjektumok és adatforrások közé, így mutatva bármely kijelölt objektum aktuális kötéseit.

A következő ábra a vizuális komponensek, kötések, adatobjektumok és adatforrások együttműködésének egyszerű áttekintése. Az ismertetőben létrehozott alkalmazás a kötések felhasználásának kicsit bonyolultabb és kreatívabb felhasználását szemlélteti. A diagram létrehozandó mintaalkalmazás kötéseit, adatobjektumait és adatforrásait nem pontosan ábrázolja.



1. ábra: Ez az ábra a vizuális komponensek, kötéskezelők, adatobjektumok és adatforrások közötti példa viszonyokat mutatja be.

Az 1. ábrán minden vizuális objektum saját kötéssel rendelkezik, amely adatobjektumhoz, illetve a gomb esetén adatforráshoz társítja. A szövegmezők kötései a mezőt az adatobjektum bizonyos tulajdonságához kötik. Az ábrán szereplő sor-adatobjektum és adatobjektum egyaránt az adatforrás szolgáltatásához intézett közvetlen hívásokból nyeri adatait. A szövegmező adatobjektuma a tábla kijelölt sorának kulcsértékét használja paraméterként a Service2 szolgáltatás hívásakor, ami a tábla kijelölt soráról feltételezhetően további információkat tartalmazó, teljes rekordot ad vissza. Ez a teljes rekord pedig a gomb művelet-kötés paramétereként lesz felhasználva a Service3 szolgáltatás hívásakor, ami lehet például a mezőbe beírt értékeket frissítő metódus. Az adatobjektumokkal, adatkötésekkel és adatforrásokkal kapcsolatos részletesebb magyarázatért használja a korábban biztosított hivatkozásokat.

Webszolgáltatás Java proxy előállítás a projektben az adott WSDL fájjal

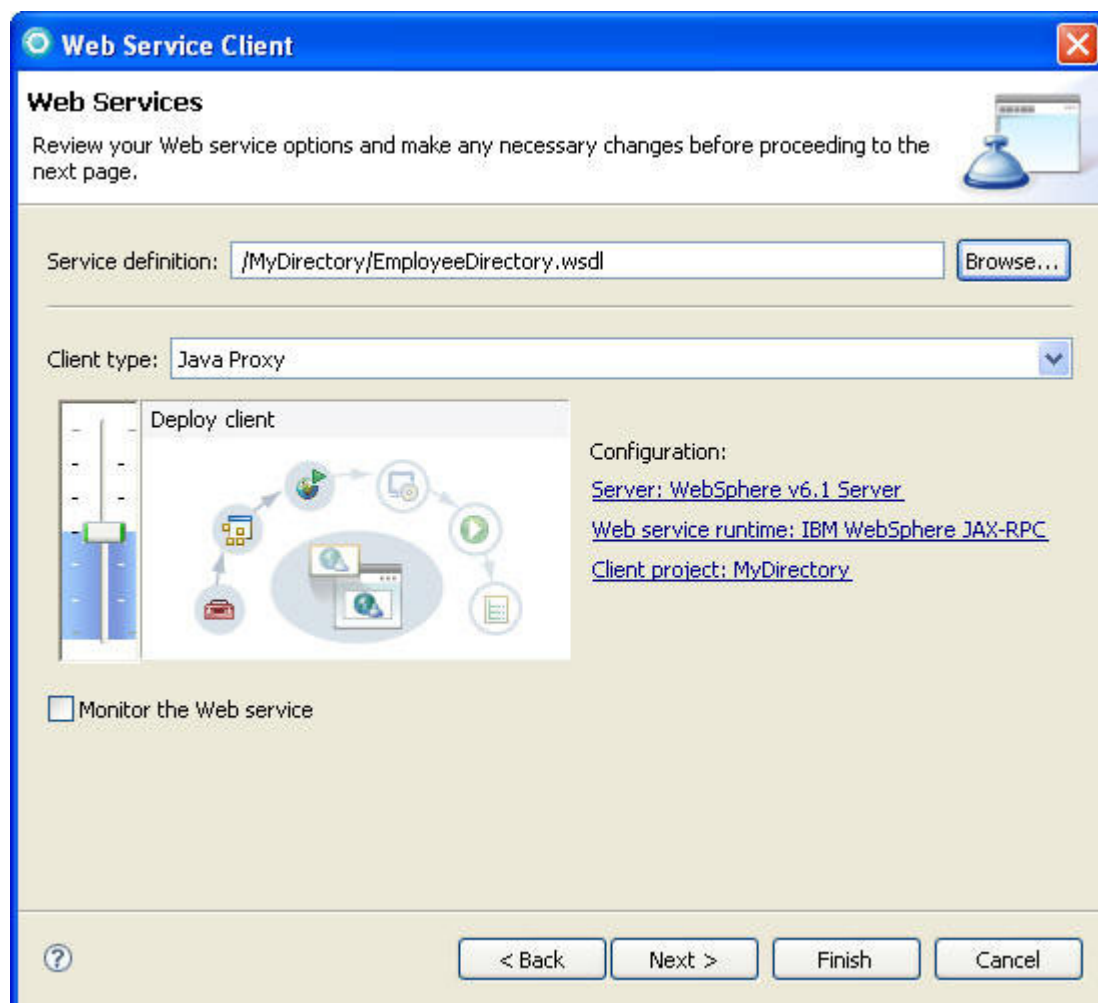
A kiszolgálón futó webszolgáltatás használatához Java proxy vagy -ügyfél szükséges a Java alkalmazással együttműködéshez. Egy WSDL fájl használatával, a Webszolgáltatás ügyfél varázsló segítségével előállíthat egy Java proxy-t a Java projektbe. A MyDirectory projekt tartalmazza az EmployeeDirectory.wsdl fájlt, amivel előállíthatja ezt a proxy-t. A Java proxy előállítása után létrehozhat egy webszolgáltatást képviselő adatforrást, és megkezdheti a vizuális összetevők összekötését.

Fontos: Az ebben a feladatban használt WSDL fájl feltételezi, hogy telepítette a webszolgáltatást a WebSphere Application Server helyi telepítésére, illetve hogy az alapértelmezett portot adta meg a saját gépnek (<http://localhost:9080>). Ha másként telepítette az EAR fájlt, akkor mielőtt folytathatná, annak megfelelően módosítania kell a WSDL fájlt.

A projekt webszolgáltatás Java proxy-jának előállításához tegye a következőket:

1. Kattintson a főmenü **Fájl** → **Új** → **Egyéb** menüpontjára, és válassza a **Webszolgáltatások** → **Webszolgáltatás ügyfél** varázslót. Ha nem jelenik meg a Webszolgáltatás kategória, akkor válassza az **Összes varázsló megjelenítése** lehetőséget.
2. Használja a varázslót a webszolgáltatás ügyfél meghatározására:

- A **Szolgáltatás meghatározás** részben adja meg a MyDirectory projektben megadott WSDL fájlt: /MyDirectory/EmployeeDirectory.wsdl
- Az **Ügyféltípus** mezőben válassza ki a **Java proxy** elemet.
- Állítsa a csúszkát az **Ügyfél telepítése** bejegyzéshez.
- Győződjön meg róla, hogy a kiszolgáló és webszolgáltatás futási környezete megfelelően be van állítva azon a kiszolgálón, amelyen a futtatást végzi. Ez az ismertető IBM WebSphere JAX-RPC futási környezettel rendelkező WebSphere 6.0 és WebSphere 6.1 termékkel került tesztelésre.
- Győződjön meg róla, hogy a MyDirectory projekt kimenete a Java proxy.



- Kattintson a **Befejezés** gombra. A Webszolgáltatás ügyfél varázsló előállítja a Java proxy-t egy új csomagban (directory.service) a projektben.

Az employeesTable webszolgáltatás által visszaadott sor adatobjektumhoz kötése

Mivel ebben az alkalmazásban az employeesTable az első kötött vizuális összetevő, létre kell hozni egy adatforrást, ami a projekt most felvett minta-webszolgáltatás proxy-jára mutat. Amikor a későbbi feladatokban más vizuális összetevőket köt, ezt az adatforrást fogja újrafelhasználni. Ebben a lépésben felveszi a Webszolgáltatás adatforrást és a lightEmployeeRecordRows adatobjektumot.

Az alkalmazottak tábla kötéséhez tegye a következőket:

- A Java komponens nézetben vagy a tervezési nézetben jelölje ki az employeesTable táblát. (Ellenőrizze, hogy nem a JScrollPane szülőösszetevőt választotta ki.) A tervezési területen megjelenik egy **Kötés** címke az employeesTable tetején.

Filter: [] Bind: [Last name]

A	B	C	D	E
0x0	0x1	0x2	0x3	0x4
1x0	1x1	1x2	1x3	1x4
2x0	2x1	2x2	2x3	2x4
3x0	3x1	3x2	3x3	3x4
4x0	4x1	4x2	4x3	4x4

New Delete

- Kattintson a **Kötés** lapra az employeesTable táblán. Ennek alternatívájaként kattintson a jobb egérgombbal az employeesTable táblára, majd válassza az előugró menü **Kötési tulajdonságok** menüpontját.
- Mivel az alkalmazásban még nincsenek adatobjektumok, fel kell venni egyet. Kattintson az **Új adatforrás adatobjektum** gombra.
- A **Forrás típusa** mezőben válassza a **Webszolgáltatás** lehetőséget.
- Mivel még nem vette fel a webszolgáltatás adatforrást az alkalmazáshoz, most kell megtennie. Kattintson az **Adatforrás** mező melletti ... gombra, hogy megnyissa a projektben elérhető webszolgáltatás ügyfeleket vagy proxykat kereső Webszolgáltatás adatforrás hozzáadása párbeszédablakot.
- Jelölje ki az EmployeeDirectory webszolgáltatás lehetőséget, és kattintson a Befejezés gombra. Egy új adatforrást adott hozzá a DirectoryApp.java fájlhoz.

Add Web Service Data Source

Web Service Data Source
Select a Web service.

Existing Web services:

Name	Class
EmployeeDirectory	directory.service.EmployeeDirectory

Add new Web service...

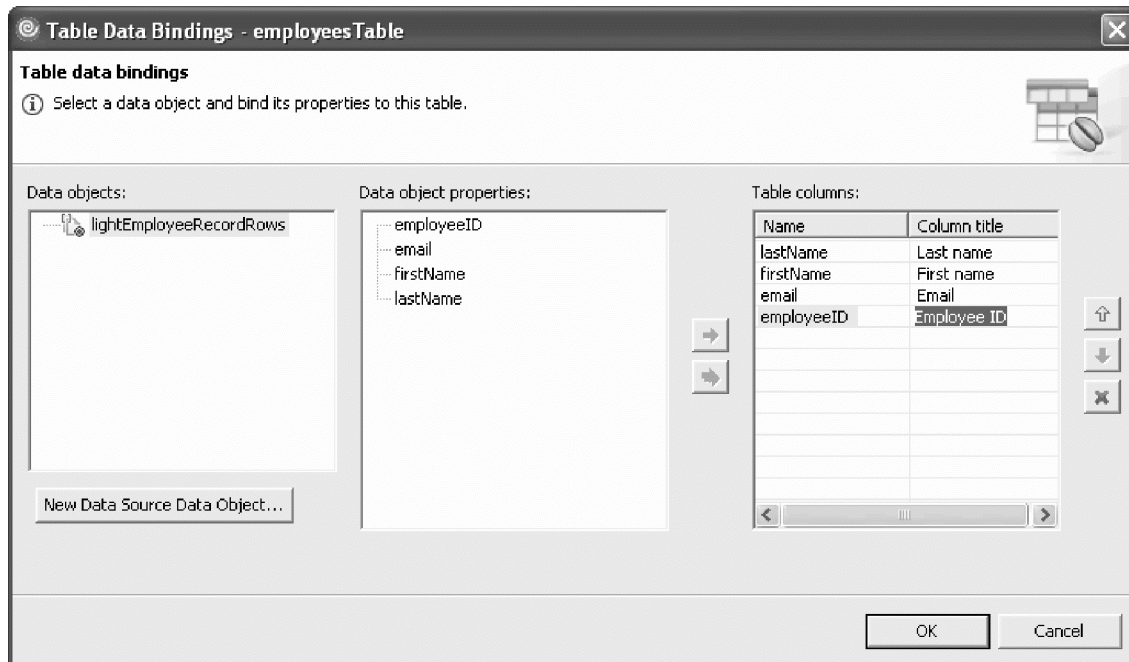
Finish Cancel

- Az Új adatforrás adatobjektum párbeszédablakban válassza a Szolgáltatásforrás mezőben a getLightEmployeeRecords() metódust, és fogadja el az új adatobjektum alapértelmezett nevét: lightEmployeeRecordRows. Ehhez a szolgáltatásmetódushoz nem szükségesek paraméterek. kattintson az OK gombra. Létrejön az új adatobjektum, és megjelenik a tervezési nézet szabadon alakítható részén.



Tipp: Mivel egy táblát köt, az Új adatforrás adatobjektum párbeszédablak csak a sor adatobjektumokat visszaadó szolgáltatásokat jeleníti meg. Ebben az esetben a getLightEmployeeRecords() metódus az egyetlen olyan elérhető szolgáltatás, ami objektumtömböt ad vissza.

8. A Tábla adatainak kötése párbeszédablakban válassza a lightEmployeeRecordRows adatobjektumot.
9. Most ki kell választania a lightEmployeeRecordRows adatobjektumnak az employeesTable táblában megjeleníteni kívánt tulajdonságait.



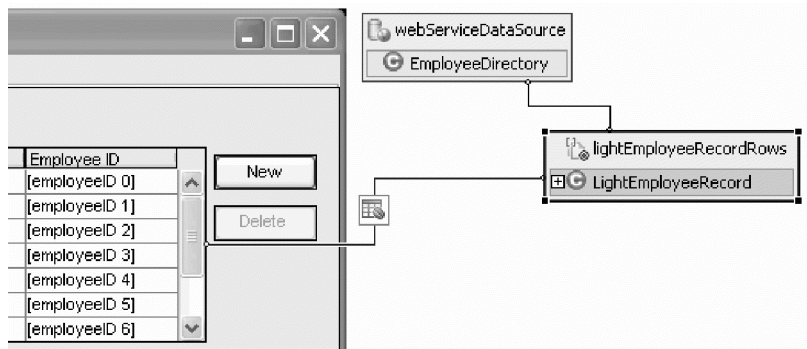
- a. Kattintson a kettős nyíl gombra, hogy az összes objektumtulajdonságot felvegye a **Tábla oszlopai** listába.
- b. A felfelé- és lefelé mutató nyilakkal rendezze át a sorokat ebbe a sorrendbe: lastName, firstName, email, employeeID

- c. Nevezze át az oszlopcímeket: Vezetéknév, Keresztnév, E-mail cím, Alkalmazott azonosítója

Tipp: A tábla kötésének befejeztével bármikor visszatérhet a kötési tulajdonságokhoz, és kívánság szerint átnevezheti, illetve átrendezheti az oszlopokat.

- d. Kattintson az **OK** gombra.

Az employeesTable már össze van kötve a lightEmployeeRecordRows adatobjektummal, JRowTableBinder összetevő használatával. Ha a szabadon alakítható területen a lightEmployeeRecordRows adatobjektumra kattint, akkor a vizuális szerkesztő egy vonalat rajzol az adatobjektumtól a tábláig. Ezen a vonalon a JRowTableBinder kötetést egy táblakötés ikon  ábrázolja. Egy másik vonal jelzi, hogy az adatobjektum adatforrása a webserviceDataSource.



Előadás ellenőrzési pont

Figyelje meg a projekt és az alkalmazás változásait. Ebben a részben felvette a webszolgáltatás adatforrást, egy sor adatobjektumot és egy kötetést, ami a sor adatobjektumhoz köti az employeesTable táblát.

Figyelje meg az új csomagot (jve.generated), ami azért lett létrehozva, hogy tárolja a vizuális Java szerkesztő által létrehozott kötési osztályokat. Figyelje meg a másik új csomagot is (directory.service), ami a webszolgáltatás Java proxy-ját tartalmazza. Írja le vagy foglalja össze az ebben a feladatban tanultakat.



Ha most futtatja a Cégcímtár alkalmazást, akkor a webszolgáltatás feltölti az alkalmazottak táblát a létező alkalmazott rekordokkal.

2.3 feladat: Részletmezők kötése a táblakijelöléshez

Az előző feladatban az employeesTable táblát kötötte a webszolgáltatás getLightEmployeeRecords() szolgáltatása által visszaadott lightEmployeeRecordRows összetevőhöz. Most a részletmezőket kell feltöltenie a táblában kijelölt alkalmazottól függően.

Az egyes kijelölt alkalmazottak extra részleteinek megszerzéséhez másik adatobjektumra lesz szükség. A következőkben felveendő selectedEmployeeRecord adatobjektumot a getFullEmployeeRecord() szolgáltatás adja vissza. Ez a szolgáltatás a táblában kijelölt alkalmazott azonosítóját használva paraméterként lehívja az alkalmazott további részleteit, például a telefonszámát és a munkahelyét.

A tábla sor adatobjektumhoz kötéséhez használt JRowTableBinder leegyszerűsíti ezt a lépést. A JRowTableBinder önálló adatobjektumként teszi ki a kijelölt elemet, amit paraméterként használhat a getFullEmployeeRecord(java.lang.Integer) metódushoz. Ezután könnyen a megfelelő tulajdonsághoz kötheti a szövegmezőket a selectedEmployeeRecord adatobjektumban.

Ismerje meg ezt a webszolgáltatást: A webszolgáltatás két olyan szolgáltatást tartalmaz, amivel megszerezheti az egyes alkalmazottak részleteit. A tábla felsorolja az összes alkalmazottat, és csak az adatok részhalmaza jelenik meg a táblában. Ezután, amikor csak egy alkalmazott van kijelölve, lekérdezheti annak a kijelölt alkalmazottnak az információit. Ha a webszolgáltatás elküldte minden alkalmazott összes adatát, amikor tábla lekérdezte azokat, akkor a webes forgalom megnőhet, az alkalmazás lassúságát okozva.

Például ha az alkalmazottak rekordja tartalmaz fényképet vagy csatolt fájlt is, akkor az összes alkalmazott lekérdezésekor nem kell lekérni az összes fényképet is. Így a `getLightEmployeeRecord` szolgáltatás feltölti a táblát, a `getFullEmployeeRecord` pedig lekérdezi a táblában kijelölt alkalmazott összes adatát.

A Vezetéknév mező kötése

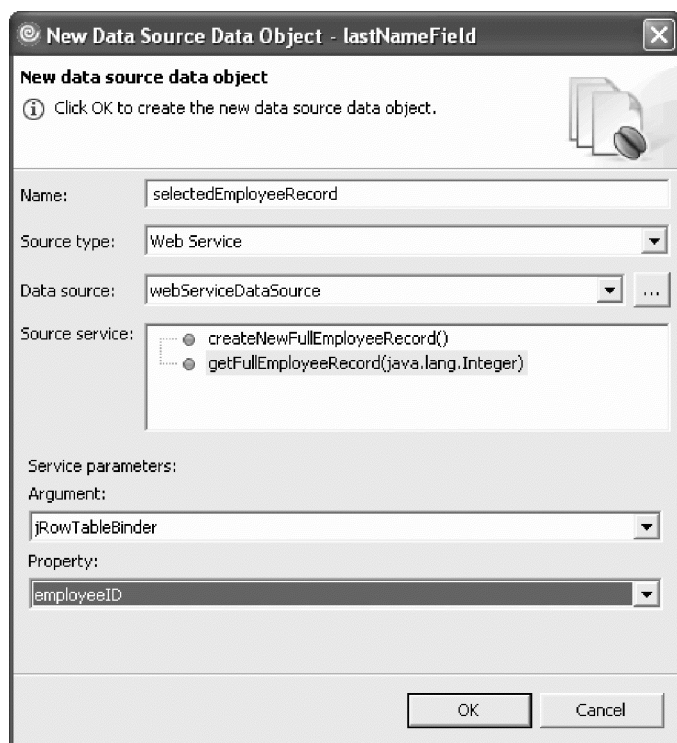
Ebben a lépésben a **Vezetéknév** mezőt köti a `selectedEmployeeRecord` adatobjektum `lastName` tulajdonságához:

1. Java komponens vagy tervezés nézetben jelölje ki a vezetéknév `TextField` mezőjét (`lastNameField`). A tervezés területen megjelenik egy **Kötés** lap a szövegmezőn.

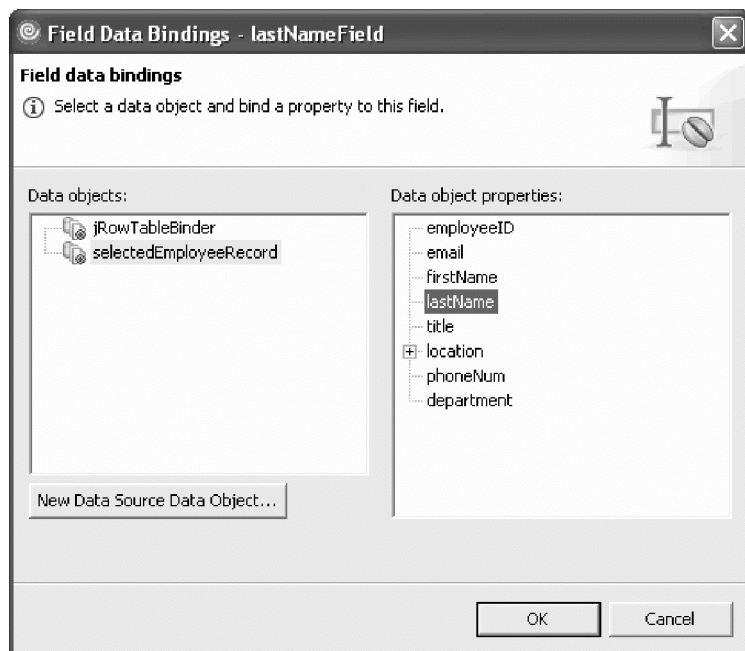


2. Kattintson a **Kötés** lapra, hogy megnyissa a Mezőadatok kötése párbeszédablakot.
3. Kattintson az **Új adatforrás adatobjektum** gombra. Bár a meglévő `jRowTableBinder` adatobjektum visszaadja a helyes vezetéknévet, nem tartalmazza a teljes alkalmazott rekordot. Új adatobjektumot kell létrehoznia, ami a teljes alkalmazotti rekordot képviseli.
4. Győződjön meg róla, hogy a **Forrás típusa** mezőben a **Webszolgáltatás** van kijelölve, illetve hogy az **Adatforrás** értéke `webServiceDataSource`.
5. Válassza a `getFullEmployeeRecord(java.lang.Integer)` szolgáltatást a **Szolgáltatásforrás** listában. Az **Új adatforrás** adatobjektum párbeszédablak felsorolja a szövegmezővel kompatibilis adatobjektumokat visszatérítő szolgáltatásokat.
6. A **Név** mezőbe írja be a `selectedEmployeeRecord` értéket.
7. A **Paraméter** mezőben válassza ki a `jRowTableBinder` összetevőt, a **Tulajdonság** mezőben pedig válassza ki az `employeeID` értéket. Így a kijelölt sor alkalmazotti azonosítója a `getFullEmployeeRecord()` szolgáltatásmetódus paramétere.

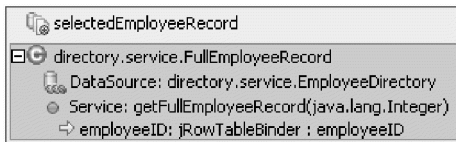
Megjegyzés: A `getFullEmployeeRecord(java.lang.Integer)` metódus paramétere csak integer lehet. A teljes rekord lekéréséhez használja az alkalmazottak tábla aktuális kijelölésében szereplő alkalmazottazonosítót. A kötött táblánál a vizuális szerkesztő automatikusan előállította a `jRowTableBinder` összetevőt, ami az alkalmazottak tábla aktuális kijelöléseit figyeli. Integer paraméterként a `jRowTableBinder` kijelölt sorának `employeeID` értékét fogja használni.



8. Kattintson az **OK** gombra.
9. Győződjön meg róla, hogy a Mezőadatok kötése párbeszédablakban a selectedEmployeeRecord van kijelölve az **Adatobjektumok** listában. Figyelje meg, hogy több tulajdonság áll rendelkezésre a selectedEmployeeRecord adatobjektumhoz, mint ahány a jRowTableBinder adatobjektumhoz.
10. Az **Adatobjektum tulajdonságai** listában jelölje ki a lastName tulajdonságot.



11. Kattintson az **OK** gombra. Az alkalmazás vezetéknév mezője már a selectedEmployeeRecord adatobjektum lastName tulajdonságához van kötve, amit a getFullEmployeeRecord() metódus ad vissza.
Létrejön egy selectedEmployeeRecord nevű új adatobjektum, és hozzáadódik az alkalmazáshoz. Az adatobjektum vizuális ábrázolása is rákerül a tervezés nézet szabadon alakítható területére, az alábbi képen láthatóan:



Ha most kijelöli a lastName mezőt a tervezési területen, akkor egy vonal jelzi, hogy a selectedEmployeeRecord objektumhoz kötött. A vonal közepén levő szöveges kötésikon jelzi a kötéshez használt SwingTextComponentBinder összetevőt. Ha a tervezési területen kijelöli a vonalat, vagy a kötést jelképező ikont, akkor a Tulajdonságok nézetben megvizsgálhatja a kötés tulajdonságait.

A hátralévő részletmezők kötése

Az alkalmazottak fennmaradó részletmezőinek kötéséhez kövesse a vezetéknévénél használt folyamatot, azt kivéve, hogy az adatobjektumot már nem kell felvennie. Mivel már felvette a selectedEmployeeRecord adatobjektumot, elég egyszerűen a megfelelő tulajdonsághoz kötni az egyes mezőket a selectedEmployeeRecord adatobjektumban.

A mezők kötéséhez tegye a következőket az alkalmazás Alkalmazottak részletei részének minden egyes mezőjéhez:

1. Válassza ki a mezőt a tervezés nézetben, és kattintson a **Kötés** lapra.
2. A Mezőadatok kötése párbeszédablakban jelölje ki a selectedEmployeeRecord objektumot az **Adatobjektumok** listából.
3. Az **Adatobjektum tulajdonságai** listában jelölje ki az adott mezőhöz tartozó tulajdonságot. Az alábbi diagram mutatja, hogy mely tulajdonsághoz kell kötni az egyes szövegmezőket:

Mező	Tulajdonság a selectedEmployeeRecord adatobjektumban
lastNameField	lastName
firstNameField	firstName
idField	employeeID
emailField	email
phoneField	phoneNum
officeField	location.office
buildingField	location.building
siteField	location.site

4. Kattintson az **OK** gombra.

A szövegmezők kötésének befejeztével a tervezési területnek a következő ábrához hasonlóan kell kinéznie:

Az alkalmazottazonosító mező írásvédetté tétele

Az alkalmazottazonosító mező tiltott, mivel a mező szerkeszthetőségi tulajdonsága hamisra van állítva. Azonban a szövegmező kötésének alapértelmezett viselkedése engedélyezett állapotra változtatja a mezőt, ha az adatobjektum tartalmaz értéket. Kikapcsolhatja ezt a kötési viselkedést, így a mező a kezdeti írásvédett állapotban marad.

Tegye a következőket, hogy megakadályozza a szerkeszthetőség tulajdonság automatikus átkapcsolását:

1. Jelölje ki az alkalmazottazonosító mezőt. A tervezési területen megjelenik egy vonal, rajta egy ikonnal , a mező kötését jelezve.
2. Kattintson az alkalmazottazonosító mező kötésének  ikonjára.
3. A Tulajdonságok nézetben módosítsa az autoEditable tulajdonságot **hamis** értékre. Nyomja meg az **Entert**.

Előadás ellenőrzési pont

Ha most futtatja az alkalmazást, és kijelöl egy alkalmazottat a táblában, akkor az alkalmazott rekordjának részletei megjelennek a részletek mezőkben.

2.4 feladat: A Frissítés gomb egy műveletgyűjtőhöz kötése

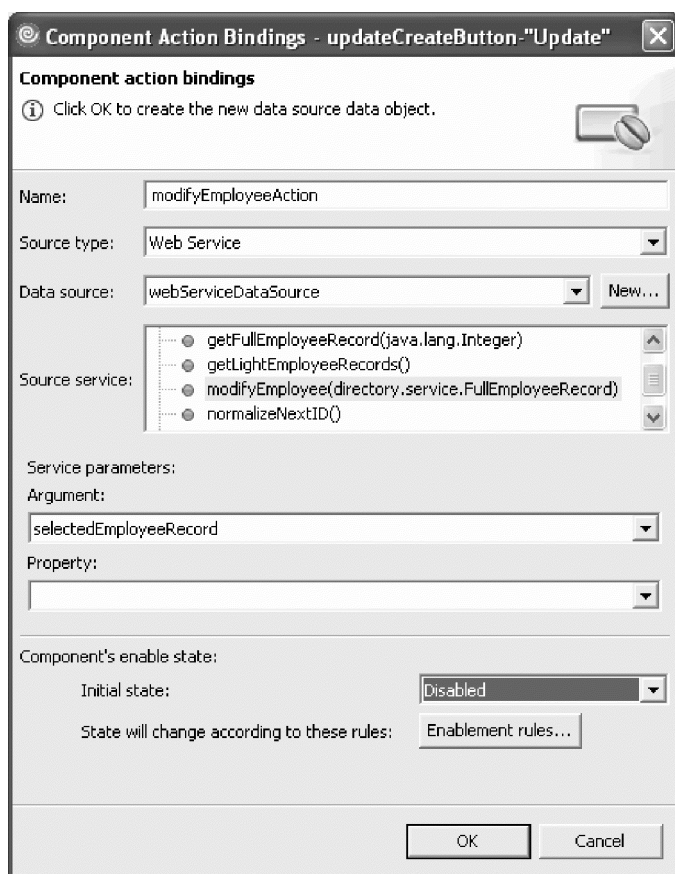
A vizuális Java szerkesztő műveleti kötésekét biztosít, egy adatforrás szolgáltatásainak egy gomb megnyomásakor meghívásához. Például amikor a Frissítés gombra kattint, az alkalmazás futtatja a webszolgáltatáson a `modifyEmployee()` metódust, a részletmezőkben megadott módosításokkal. Ebben a részben a Frissítés gombot fogja művelet kötéshez kötni.

A Frissítés gomb kötéséhez tegye a következőket:

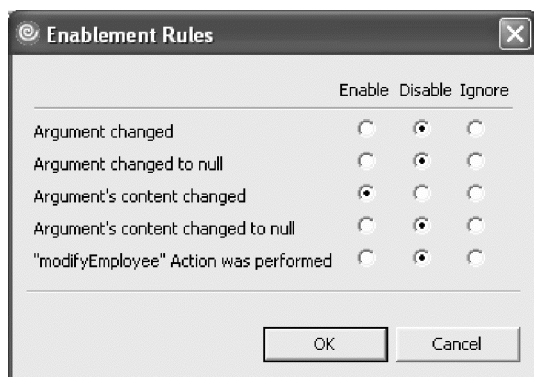
1. Jelölje ki a tervezési területen a **Frissítés** gombot, és kattintson a **Kötés** lapra az Összetevő műveleti kötései párbeszédablak megnyitásához.



2. A **Forrás típusa** mezőben válassza a **Webszolgáltatás** lehetőséget.
3. Az **Adatforrás** mezőben válassza a **webServiceDataSource** projektet.
4. Jelölje ki a **Szolgáltatásforrás** listából a **modifyEmployee(directory.service.FullEmployeeRecord)** szolgáltatást.
5. A **Név** mező automatikusan **modifyEmployeeAction** értékre változik. Fogadja el ezt az alapértelmezett értéket.
6. A **Paraméter** mezőben jelölje ki a **selectedEmployeeRecord** értéket.
7. Mivel a `modifyEmployee()` metódus egy teljes alkalmazott rekordot vesz paraméteréül, a **Tulajdonság** mezőt üresen kell hagyni.
8. Állítsa a gomb **Kezdeti állapot** értékét **Tiltott** értékre.



9. A gomb állapotváltozásainak meghatározásához kattintson a **Felkészítési szabályok** gombra. Adja meg, hogy a gomb csak akkor legyen engedélyezett, amikor a paraméter tartalma módosul, minden más esetben maradjon tiltott. Kattintson az **OK** gombra.



Ez azt jelenti, hogy a **Frissítés** gomb tiltott marad mindaddig, míg a selectedEmployeeRecord meg nem változik. Más szavakkal, amint új értéket ír be az egyik, a selectedEmployeeRecord objektumhoz kötött részletmezőbe, a kötés engedélyezi a gombot. Ha másik rekordot jelöl ki, vagy a **Frissítés** gombra kattint, akkor a gomb ismét tiltott lesz.

10. Kattintson az **OK** gombra.

Egy új SwingDataServiceAction kötés került a **Frissítés** gombhoz. Ha kijelöli a gombot a tervezési területen, akkor a vizuális szerkesztő egy vonalat rajzol, ami jelzi, hogy a gomb a webszolgáltatás adatforráshoz van kötve. Rózsaszín pontozott nyíl mutat a selectedEmployeeRecord objektumtól a vonalra. Ez a nyíl jelzi, hogy a szolgáltatás meghívásának paramétere a selectedEmployeeRecord objektum.

Előadás ellenőrzési pont

Az alkalmazás futtatásakor most már frissítheti egy alkalmazott rekordját.

Jelöljön ki egy alkalmazottat a táblában, és írja át a vezetéknévét. Amint módosítja a vezetéknévet, a **Frissítés** gomb engedélyezett lesz. Amikor a **Frissítés** gombra kattint, meghívja a `modifyEmployee` szolgáltatást, és az alkalmazott rekordja frissül. Az új vezetéknév látszódik az alkalmazottak táblában.

2.5 feladat: A Törlés gomb és a megerősítés párbeszédablak engedélyezése

Ebben a részben egy alkalmazott rekord törlésére programozza be a Cégcímtár alkalmazást.

Az alábbi lista írja le az alkalmazás által követendő viselkedést:

- A tábla egy alkalmazott rekordjának kijelölésekor a **Törlés** gomb engedélyezetté válik.
- A **Törlés** gombra kattintáskor megjelenik a Törlés megerősítése párbeszédablak, és kéri a törlés megerősítését.
- Ha a Törlés megerősítése párbeszédablak **Igen** gombjára kattint, akkor az alkalmazott rekord törlődik, bezáródik a Törlés megerősítése párbeszédablak, és frissül az alkalmazottak listája.
- Ha a **Nem** gombra kattint, akkor a törlés félbeszakad, és bezárul a Törlés megerősítése párbeszédablak.

A Törlés gomb engedélyezett vagy tiltott állapotának programozása annak alapján, hogy van-e kijelölt sor a táblában

A Törlés gomb engedélyezett vagy tiltott állapotának programozásához vegyen fel egy figyelőt a táblába, ami engedélyezi a gombot, ha van kijelölt sor.

1. Jelölje ki az `employeesTable` táblát a Java komponens nézetben. A forrás nézet kiemeli az alábbi sort:

```
employeesTable = new JTable();
```

2. Közvetlenül ezután a sor után vegyen fel egy új `ListSelectionListener` és `valueChanged` eseményt az `employeesTable` táblába:

```
employeesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        getDeleteButton().setEnabled(getEmployeesTable().getSelectedRowCount() != 0);  
    }  
});
```

3. A kódsorok hozzáadása után a forrásszerkesztő hibásnak jelöli őket, amíg nem importálja a `ListSelectListener` és `ListSelectionEvent` eseményt. A szükséges importok hozzáadásához kattintson a **Forrás** → **Importok szervezése** menüpontra a főmenüben. A következő sorok hozzáadódnak az osztály import részéhez:

```
import javax.swing.event.ListSelectionEvent;  
import javax.swing.event.ListSelectionListener;
```

Mostantól ha a tábla egy sora ki van jelölve, akkor a **Törlés** gomb engedélyezett.

A Törlés megerősítése párbeszédablak megnyitása a Törlés gombra kattintáskor

Vegyen fel egy `actionPerformed` eseményt a Törlés gombhoz, és programozza az eseményt a Törlés megerősítése párbeszédablak megnyitására.

1. Kattintson a jobb egérgombbal a **Törlés** gombra, majd válassza az előugró menü **Események** → **actionPerformed** menüpontját. Az alábbi eseménycsont lesz hozzáadva a `getDeleteButton()` metódushoz:

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        System.out.println("actionPerformed()");  
        // TODO automatikusan előállított eseménycsont actionPerformed()  
    }  
});
```

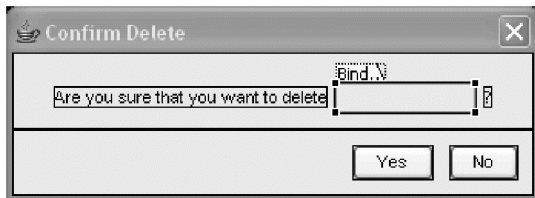
2. Cserélje ki ezt az előállított csontot a következő kóddal, ami láthatóvá teszi a Törlés megerősítése párbeszédablakot a gombra kattintáskor:

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        getConfirmDialog().setVisible(true);  
    }  
});
```

A Törlés megerősítése párbeszédablak szövegmezőjének kötése

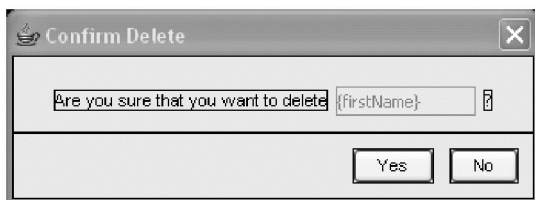
Kösse a Törlés megerősítése párbeszédablak szövegmezőjét úgy, hogy a törölni kívánt alkalmazott keresztnévét jelenítse meg.

1. A Java komponens nézetben vagy a tervezési területen jelölje ki az `employeeToDeleteField` szövegmezőt, és kattintson a **Kötés** lapra.



2. A mezőadatok kötése párbeszédablakban jelölje ki a `selectedEmployeeRecord` adatobjektumot és a `firstName` mezőt, majd kattintson az **OK** gombra.

A szövegmező már az `employeesTable` kijelölt sorának `firstName` oszlopához van kötve.



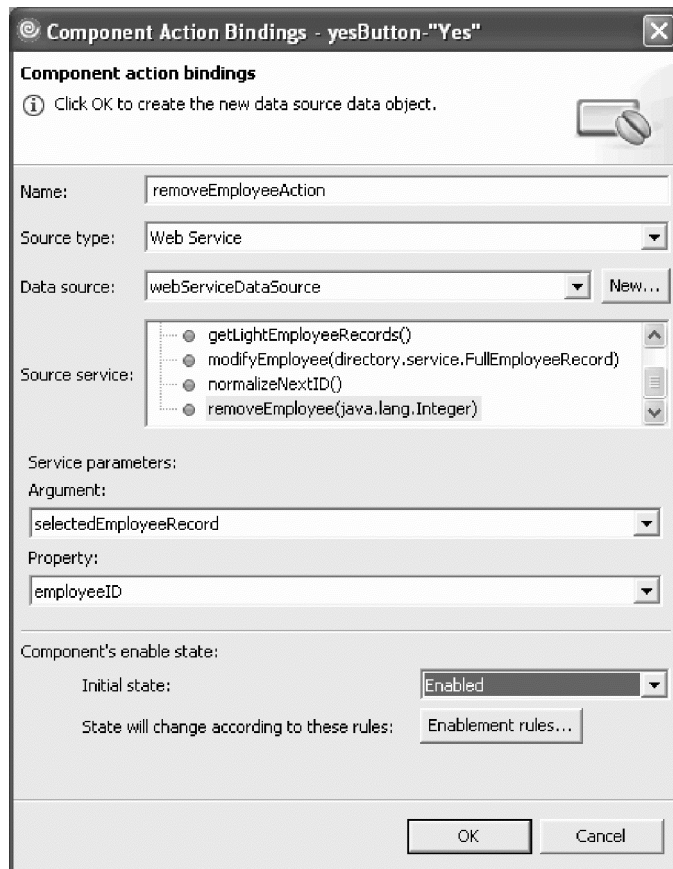
3. A mező írásvédettségének megerősítéséhez állítsa a mező kötésének **autoEditable** tulajdonságát **hamis** értékre.

Az Igen gomb kötése a törlés végrehajtásához

Kösse az **Igen** gombot a webszolgáltatás `removeEmployee(java.lang.Integer)` módszerának meghívásához.

1. Jelölje ki az **Igen** gombot, és kattintson a **Kötés** lapra az Összetevő műveleti kötései párbeszédablak megnyitásához.
2. A **Forrás típusa** mezőben válassza a **Webszolgáltatás** lehetőséget.
3. Az **Adatforrás** mezőben válassza a **webServiceDataSource** projektet.
4. A **Szolgáltatásforrás** listában jelölje ki a **removeEmployee(java.lang.Integer)** metódust.
5. A **Név** mező automatikusan **removeEmployeeAction** értékre változik. Fogadja el ezt az alapértelmezett értéket.
6. A **Paraméter** mezőben jelölje ki a **selectedEmployeeRecord** értéket.
7. A **Tulajdonság** mezőben jelölje ki a **employeeID** értéket. Mivel a `removeEmployee()` metódus integert használ paraméterként, a `selectedEmployeeRecord` alkalmazotti azonosítóját fogja használni.
8. Állítsa a gomb **Kezdeti állapot** értékét **Engedélyezett** értékre.
9. A **Felkészítés szabály** értékeiként állítsa be az összes feltételt **Mellőz** értékre.

Ez az összetevő állapot azt jelenti, hogy az Igen gomb mindig engedélyezett, mivel nincs szükség az állapot módosítására.



10. Kattintson az **OK** gombra.

Esemény felvétele a Törlés megerősítése párbeszédablak elrejtésére az alkalmazott törlése után

Ebben a lépésben egy eseményt ad az **Igen** gomb *kötéséhez* (nem magához az **Igen** gombhoz). Be szeretné zárni a Törlés megerősítése párbeszédablakot az alkalmazott törlése után, azaz miután a kötés sikeresen meghívta a szolgáltatást az adatforrásra.

Adja hozzá az alábbi kódot a `getRemoveEmployeeAction()` metódushoz:

```
removeEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        getConfirmDialog().setVisible(false);
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

Ez az eseménykód elrejt a Törlés megerősítése párbeszédablakot, miután befejeződött a kötés művelete.

Előadás ellenőrzési pont

A Cégcímtár alkalmazás futtatásakor most már kijelölhet egy alkalmazottat a táblában, rákattinthat a **Törlés** gombra, majd az **Igen** gombra a törlés megerősítéséhez. Az alkalmazott rekordja el lesz távolítva a címtárból, és az alkalmazottak listája tükrözni fogja ezt a változást.

2.6 feladat: Műveletek és kötések beállítása új alkalmazott felvételéhez

Ebben a részben lehetővé teszi új alkalmazott rekord létrehozását a Cégcímtár alkalmazásban.

Mivel az alkalmazás viselkedése bonyolultabb és dinamikusabb új alkalmazott hozzáadásánál, ez a feladat is bonyolultabb, és megkívánja a forráskód kézi módosítását. Ez a feladat bemutatja továbbá az adatobjektumok egyes fejlettebb képességeit is, és kreatív példákkal mutatja be a kötések és adatobjektumok egyéni célokra használásának módjait.

Az alábbi lista az alkalmazás kívánt viselkedését írja le:

- Amikor az **Új** gombra kattint, a következő viselkedés várható el:
 - Megszűnik a kijelölés az alkalmazottak táblában, és a tábla tiltott lesz.
 - A tábla kijelölésének megszűnése miatt a **Törlés** gomb is tiltott lesz.
 - A **Szűrő** mező tiltott lesz.
 - A részletmezőkből törlődik minden érték, kivéve az új alkalmazottazonosítót.
 - A **Frissítés** gomb szövege **Hozzáadás** értékre változik.
- Amikor a **Hozzáadás** gombra kattint, a következő viselkedés várható el:
 - A részletmezőkben megadott értékek új alkalmazott rekordként hozzáadódnak a címtárhoz.
 - A tábla engedélyezett lesz, és frissülnek az értékek.
 - A **Szűrő** mező engedélyezett lesz.
 - A **Hozzáadás** gomb szövege visszaváltozik **Frissítés** értékre.

Új adatforrás adatobjektum hozzáadása a createNewFullEmployeeRecord() metódus meghívására

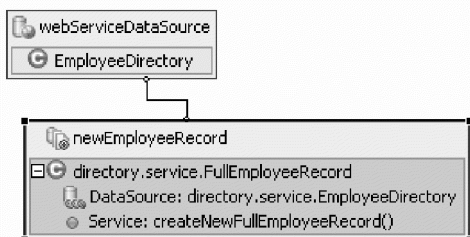
A minta-webszolgáltatás nyújtotta createNewFullEmployeeRecord szolgáltatás új, üres alkalmazott rekordot biztosít, kitöltve a következő rendelkezésre álló alkalmazottazonosítóval. Ezt az üres rekordot lehet azután feltölteni az új alkalmazott információival, illetve visszaküldeni a webszolgáltatásnak.

1. A vizuális Java szerkesztő palettáján bontsa ki az Adatobjektumok fiókot, és válassza az **Adatforrás adatobjektum** lehetőséget.
2. Vigye a tervezési terület üres része fölé az egérmutatót, és kattintson a bal egérgombbal az Adatforrás adatobjektum lerakásához. Az új Adatforrás adatobjektum hozzáadódik és megjelenik a szabadon alakítható területen:



3. Kattintson a jobb egérgombbal az Adatforrás adatobjektum összetevőre, majd válassza az előugró menü **Mező átnevezése** menüpontját. Nevezze át az adatobjektumot newEmployeeRecord névre.
4. Kattintson a jobb egérgombbal a newEmployeeRecord adatobjektumra, majd válassza az előugró menü **Kötési tulajdonságok** menüpontját. Megjelenik az Adatok kötése párbeszédablak.
5. Az **Adatforrás** mezőben válassza a webServiceDataSource projektet.
6. A **Szolgáltatás** mezőben válassza a createNewFullEmployeeRecord() metódust.
7. Kattintson az **OK** gombra.

A szabadon alakítható területen látható, hogy a newEmployeeRecord adatforrás adatobjektum a webszolgáltatáshoz kötött.



Alapszintű adatobjektum hozzáadása az adatobjektumok átkapcsolásának kihasználására

Mivel a részletmezőknek és a Frissítés gombnak módot kell váltaniuk (frissítés elvégzéséhez és új alkalmazott létrehozásához), különböző időpontokban különböző adatobjektumhoz kell kötni őket. A lépés elvégzéséhez hozzá kell adni egy switchingDataObject nevű Alapszintű adatobjektumot. Ezt az Alapszintű adatobjektumot fogja a szövegmezők kötésének váltogatására használni a selectedEmployeeRecord és az newEmployeeRecord objektum között.

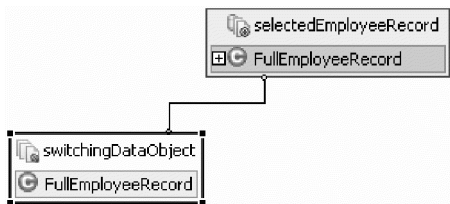
Az új Alapszintű adatobjektum egyszerűen egy másik adatobjektumra mutat (selectedEmployeeRecord), amit az egyik előző feladatban adott meg. Ez az új adatobjektum hasznos lesz, amikor olyan metódust hoz létre, ami ezt az alapszintű adatobjektumot a korábban létrehozott newEmployeeRecord használatára utasítja. Más szavakkal, ez az alapszintű adatobjektum köztes adatobjektum funkcióként fog szolgálni, ami a selectedEmployeeRecord adatobjektum és a newEmployeeRecord adatobjektum közt kapcsol át, lehetővé téve az alkalmazás vizuális összetevőinek, hogy két különböző adatobjektummal dolgozzanak.

1. A vizuális szerkesztő palettán válassza ki az **Alapszintű adatobjektum** elemet, és rakja a szabadon alakítható területre. Hozzáadódik egy basicDataObject.



2. Nevezze át az adatobjektumot switchingDataObject
3. A switchingDataObject tulajdonságok nézetében állítsa a **sourceObject** tulajdonságot **selectedEmployeeRecord** értékre. A selectedEmployeeRecord értéket a tulajdonság Érték oszlopának legördülő menüjéből választhatja ki.

A switchingDataObject így a selectedEmployeeRecord objektumra hivatkozik, és ugyanazokat az értékeket mutatja:

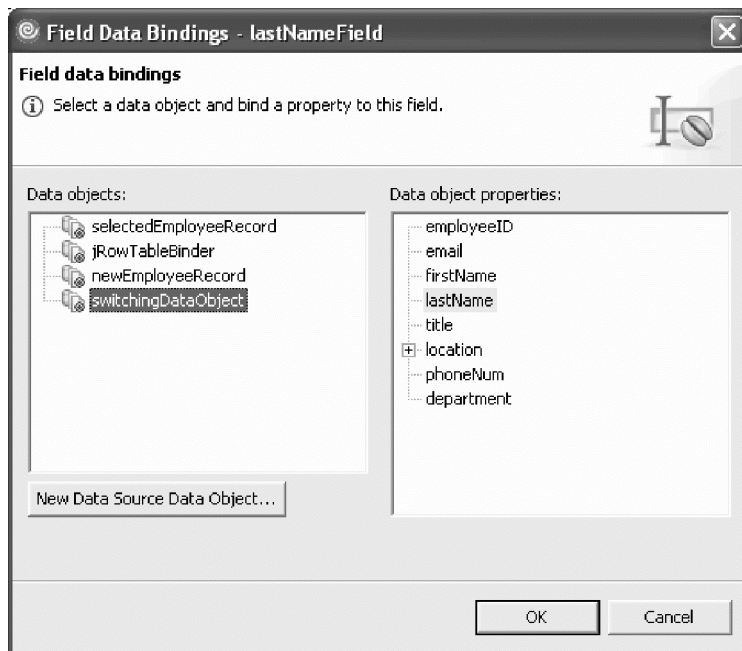


Minden alkalmazott mező átkötése a switchingDataObject objektumhoz

Bár minden alkalmazott részletmező hozzá van már kötve a selectedEmployeeRecord objektumhoz, most a switchingDataObject objektumhoz kell kötni őket. A mezők kötése után dinamikusan válthat a mezők adatobjektumai közt attól függően, hogy létező alkalmazottrekordot módosít vagy új alkalmazott rekordot vesz fel.

Az Alkalmazottak részletei rész minden mezőjénél végezze el a következőket:

1. Jelölje ki a mezőt és kattintson a **Kötés** lapra.
2. A Mezőadatok kötése párbeszédablakban válassza ki a switchingDataObject objektumot. Korábban a selectedEmployeeRecord objektumhoz kötötte a mezőket.



- Ellenőrizze, hogy a mező még mindig a megfelelő adatobjektum tulajdonságához van kötve, majd kattintson az **OK** gombra. Ha kijelöli a mezőt a tervezési nézetben, akkor látható, hogy a kötési vonalak már a switchingDataObject objektumra mutatnak.



Jelző és metódus meghatározása módok frissítéséhez és váltásához

Az alábbi updateMode() metódus ellenőrzi, hogy a módjelző újra van-e állítva, és ennek megfelelően módosítja az alkalmazás viselkedését. Az isNewMode logikai jelző alapértelmezésben hamisra van állítva, és az updateMode() metódus engedélyezi az alkalmazott táblát, illetve a szűrő mezőt, valamint "Frissítés" értékre változtatja a Frissítés gomb feliratát. Ha az isNewMode igazra van állítva, akkor az alkalmazottak tábla tiltott, eltűnik róla minden kijelölés, a szűrő mező tiltott lesz, valamint a Frissítés gomb felirata "Hozzáadás" szövegre változik.

Adja hozzá a következő kódot a DirectoryApp.java osztályhoz, az utolsó bezáró kapcsos zárójel elé:

```
private boolean isNewMode = false;
private void updateMode() {
    if (isNewMode) {
        getEmployeesTable().clearSelection();
        getEmployeesTable().setEnabled(false);
        getFilterField().setEditable(false);
        getUpdateCreateButton().setText("Add");
    } else {
        getEmployeesTable().setEnabled(true);
        getFilterField().setEditable(true);
        getUpdateCreateButton().setText("Update");
    }
}
```

ActionPerformed esemény hozzáadása az Új gombhoz

Ebben a lépésben eseménykódot vesz fel az Új gomb kattintásához. Az esemény a newEmployeeRecord használatára utasítja a switchingDataObject objektumot, átállítja a módjelzőt "új" értékre, és futtatja az előző lépésben felvett updateMode() metódust.

- Kattintson a jobb egérgombbal a tervezési nézetben az Új gombra, majd válassza az előugró menü **Események** → **actionPerformed** menüpontját. Az alábbi kód jön létre a getNewButton() metódusban:

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()");
        // TODO automatikusan előállított eseménycsonk actionPerformed()
    }
});
```

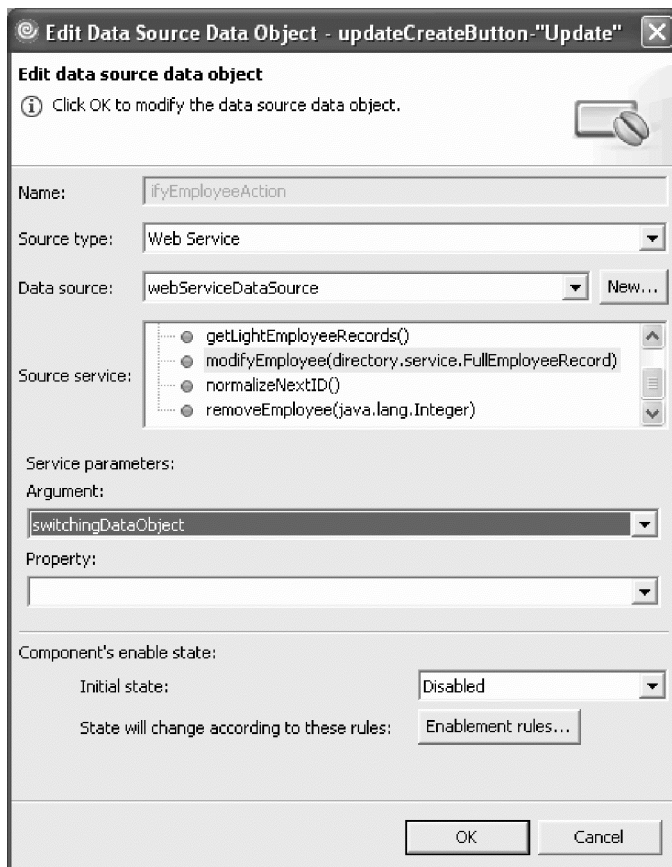
2. Cserélje ki ezt az előállított csonkot az alábbi kódra:

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getSwitchingDataObject().setSourceObject(getNewEmployeeRecord());
        getNewEmployeeRecord().refresh();
        isNewMode = true; //sets application to new mode
        updateMode(); //changes UI according to new mode
        getLastNameField().grabFocus();
    }
});
```

A Frissítés gomb újrakötése

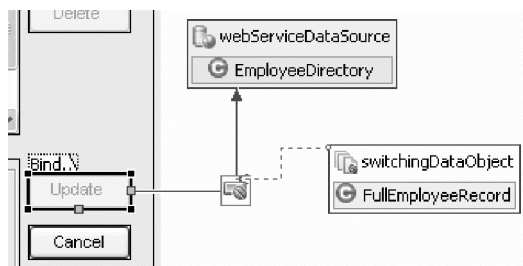
Az előző részben beprogramozta a **Frissítés** gombot a webszolgáltatás modifyEmployee metódusának használatára. A művelet SwingDataServiceAction műveletként lett megvalósítva. A SwingDataServiceAction egyik tulajdonsága a forrásobjektum, ami a szolgáltatás paramétereként viselkedik. A módosítás művelet forrásobjektuma jelenleg a selectedEmployeeRecord. Ahhoz, hogy a gombot a frissítés és a hozzáadás vezérlésére is programozhassa, újra kell konfigurálnia a gomb műveleteit, hogy a switchingDataObject objektumot használja a modifyEmployee szolgáltatás paramétereként.

1. A tervezés nézetben kattintson a **Frissítés** gombra. Figyelje meg a rózsaszín pontozott nyilat, ami jelzi, hogy a szolgáltatás hívásának paramétere a selectedEmployeeRecord.
2. Kattintson a **Kötés** lapra a **Frissítés** gombon.
3. A **Paraméter** mezőben válassza ki a switchingDataObject objektumot.



4. Kattintson az **OK** gombra.

Figyelje meg, hogy a gomb akciója úgy van konfigurálva, hogy a switchingDataObject objektumot használja paramétereként a modifyEmployee metódushoz:



Esemény hozzáadása a Frissítés gomb kötéséhez a mód visszaállításához

Miután a **Frissítés** gombra kattintott, és befejeződött a webszolgáltatás művelete, az alkalmazásnak vissza kell térnie az alapértelmezett módba és viselkedéshez. Ehhez egy eseményfigyelőt fog hozzáadni a gomb műveleti kötéséhez, ami a frissítés vagy hozzáadás művelet elvégzése után frissíti a módot és a táblát.

Adja hozzá az alábbi kódot a Frissítés gomb getModifyEmployeeAction() metódusához:

```
modifyEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        if (isNewMode) {
            //Go back to using the selectedEmployeeRecord
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            //Revert out of new mode
            isNewMode = false;
            updateMode();
        }
        // Refresh the table's data object
        getLightEmployeeRecordRows().refresh();
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

Előadás ellenőrzési pont

A Cégcímár futtatásakor most már kattinthat az **Új** gombra, és új alkalmazott rekordot vehet fel.

2.7 feladat: A Mégse gomb viselkedésének programozása

Az alkalmazás használatakor képesnek kell lennie az alkalmazott rekordok bármilyen módosításának visszavonására, ha úgy dönt, hogy mégsem küldi el a változtatásokat. Más szavakkal, képesnek kell lennie a mezők tartalmának kitorlérésére, hogy újakezdhesse a változtatások bevitelét. A működés hozzáadásához be kell állítani néhány actionPerformed eseményt a **Mégse** gombon.

A következő lista írja le a **Mégse** gomb kívánt viselkedését:

- Amikor a **Mégse** gombra kattint új módban, az alkalmazás visszaáll az új módból.
- Ha egy alkalmazott rekord módosítása közben kattint a **Mégse** gombra, akkor minden módosított érték visszatér az eredeti értékre.

Ahhoz, hogy a kívánt viselkedés eléréséhez actionPerformed eseményt vegyen fel a **Mégse** gombhoz, tegye a következőket:

1. Kattintson a jobb egérgombbal a tervezési nézetben a **Mégse** gombra, majd válassza az előugró menü **Események** → **actionPerformed** menüpontját. Az alábbi kód jön létre a getCancelButton() metódusban:

```
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()");
        // TODO automatikusan előállított eseménycsonk actionPerformed()
    }
});
```

2. Cserélje ki az előállított eseménycsonkot az alábbi kódra:


```

cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        if (isNewMode) {
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            isNewMode = false;
            updateMode();
        } else {
            getSelectedEmployeeRecord().refresh();
        }
    }
});

```

Előadás ellenőrzési pont

Ebben a részben megtanulta, hogyan lehet actionPerformed eseményt programozni a **Mégse** gombhoz.

2.8 feladat: Szűrő beállítása az alkalmazottak táblára

A Szövegszűrő kötéssel szűrheti az alkalmazottak tábla tartalmát. A szűrő egy szövegmezőből veszi a bemenetet, és a tábla egy bizonyos tulajdonsága vagy oszlopa alapján szűri a táblát.

Az alkalmazásban a **Szűrő** mezőbe írt karakterekkel fog az alkalmazott vezetékneve alapján szűrni. Ha az alkalmazott rekordjának vezetéknevében szerepelnek pontosan azok az értékek, amiket a **Szűrő** mezőbe írt, akkor az az alkalmazott rekord megjelenik a táblában.

Filter: (Last name)

Last name	First name	Email	Employee ID
Maxwell	Seth	seth.maxwell@my...	24561
Maxwell	Aubrey	aubrey.maxwell@...	30089
Martinez	James	james.martinez@m...	31780

New

Delete

Tegye a következőket a tábla szűrőjének létrehozásához:

1. Kattintson az employeesTable kötéskezelő ikonjára, majd válassza a **Szűrő kötés tulajdonságai** elemet. Megjelenik a Szűrő kötése párbeszédablak.
2. A **Szűrő bemenet szövegmezője** listában jelölje ki a **filterField** elemet.
3. A **Tábla szűrt tulajdonsága** listában jelölje ki a **lastName** elemet.

Filter Binding - employeesTable

Filter binding
Configure a binder that filters the table entries based on user input in the text field.

Text field for the filter input:
filterField

Table property to be filtered:
employeeID
email
firstName
lastName

OK Cancel

4. Kattintson az **OK** gombra.

Új SwingPropertyFilter jön létre. A tábla kötésének szűrő tulajdonsága ezt az új szűrőt fogja használni. Az új szűrő konfigurálása szerint a **Szűrő** mezőt használja bemenetként, és a tábla lastName tulajdonsága alapján szűr.

Előadás ellenőrzési pont

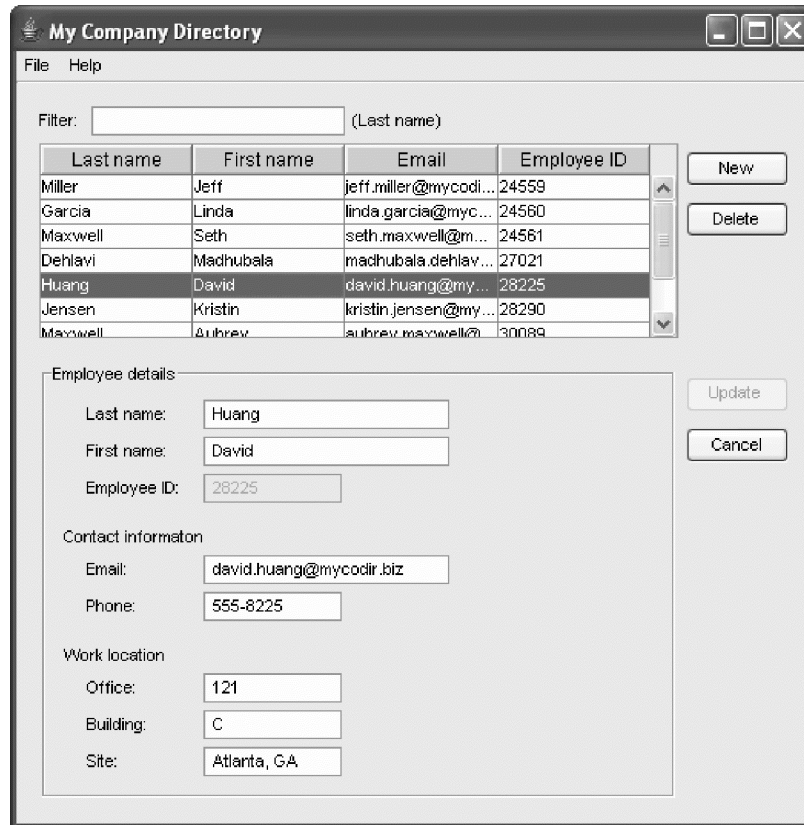
Ebben a részben megtanulta, hogyan kell szűrőt beállítani a táblához.

A Cégcímtár alkalmazás futtatásakor most már beírhat karaktereket a **Szűrő** mezőbe, és a tábla szűrése után csak azok a sorok jelennek meg, amelyek vezetéknev értéke tartalmazza a megadott karaktereket.

Gratulálunk! Elkészítette a Cégcímtár alkalmazást.

Összegzés: Funkciógazdag, webszolgáltatást használó Java ügyfél összeállítása

Gratulálunk! Megtanulta a vizuális Java szerkesztő használatát, amivel felépítheti a Cégcímtár alkalmazást, egy funkciógazdag Java ügyfelet, ami egy minta-webszolgáltatáshoz csatlakozva tartja karban az alkalmazottak címtárát.



A kész termék képe:

Tanultak

A vizuális szerkesztőt használta a grafikus felhasználói felület befejezéséhez, a GridBagLayout összetevőt használva az alkalmazott tábla elrendezéséhez. Ezután a megfelelő adatobjektumokhoz és adatforrásokhoz kötötte a táblát, a mezőket és gombokat, hogy az alkalmazás kezelni tudja az előállított webszolgáltatás Java proxy-t. Összetett kódolást is végzett, hogy az alkalmazás megfelelően, könnyen kezelhetően és intuitívan viselkedjen. Ezen kívül azt is megtanulta, hogyan lehet vállalati alkalmazást telepíteni a WebSphere Application Server 6.0 változatára, illetve hogyan lehet webszolgáltatást telepíteni.

A legfontosabb azonban az, hogy megismerte a szövegszerkesztő Java szerkesztő által nyújtott hatékony adatkezelést biztosító kötési osztályokat. Készen áll az önálló kísérletezésre, hogy új, izgalmas felhasználási módot találjon a kötésekhez.

Már készen áll az alábbi feladatok elvégzésére:

- A vizuális Java szerkesztő használata az összetevők GridBagLayout sémával elrendezésére.
- Vizuális osztály Java komponensként futtatása.
- Java alkalmazás vizuális összetevőinek webszolgáltatás által visszaadott metódusokhoz és adatobjektumokhoz kötése.
- Események felvétele vizuális összetevőkhöz.

További információforrások

Címtár alkalmazás kész verziójának importálása

Ez a projekt a kész alkalmazást, a jve.generated csomagot tartalmazza a kötéskezelő osztályokkal, valamint a Webszolgáltatás Java ügynököt beállítva a WebSphere Application Server 6.1 alkalmazáskiszolgálóra. Ha a projektet úgy importálja, hogy előtte nem végezte el az ismertető lépéseit, akkor elképzelhető, hogy be kell állítania a Java összeépítési elérési út változót. A WebSphere 6.1 webszolgáltatások egyszerűsített ügyfél JAR fájljára kell mutatnia.

Tipp: Ha az importálás során nem ad meg másik projektnevet, akkor felülírja a MyDirectory projekt tartalmát.