



Budowanie pełnego klienta Java korzystającego z usługi Web Service

Spis treści

Budowanie pełnego klienta Java korzystającego z usługi Web Service . . . 1

Wprowadzenie: Budowanie pełnego klienta Java korzystającego z usługi Web Service.	1
Moduł 1: Projektowanie interfejsu GUI klienta w edytorze wizualnym	3
Lekcja 1.1: Konfigurowanie projektu Java	3
Lekcja 1.2: Dodanie tabeli pracowników i określenie jej układu.	4
Lekcja 1.3: Uruchomienie klasy wizualnej	8
Moduł 2: Wiązanie elementów wizualnych z usługą Web Service	10
Lekcja 2.1: Zainstalowanie i wdrożenie usługi Web Service	10
Lekcja 2.2: Powiązanie tabeli pracowników ze źródłem danych usługi Web Service	12

Lekcja 2.3: Powiązanie pól szczegółów z wyborem w tabeli.	17
Lekcja 2.4: Powiązanie przycisku Aktualizacja z konsolidatorem działań	21
Lekcja 2.5: Włączenie przycisku Usuń i okna dialogowego potwierdzenia	23
Lekcja 2.6: Skonfigurowanie działań i powiązań w celu dodania nowego pracownika.	25
Lekcja 2.7: Programowanie działania przycisku Anuluj	30
Lekcja 2.8: Skonfigurowanie filtru dla tabeli pracowników	31
Podsumowanie: Budowanie pełnego klienta Java korzystającego z usługi Web Service	33

Budowanie pełnego klienta Java korzystającego z usługi Web Service

Ten kurs przedstawia sposób korzystania z wizualnego edytora Java w celu zbudowania pełnego klienta Java łączącego się z usługą Web Service. Klient budowany w ramach kursu ma nazwę Firmowa książka adresowa.

Aplikacja Firmowa książka adresowa jest aplikacją Java używaną do obsługi katalogu pracowników firmy. Aplikacja ta łączy się z przykładową usługą Web Service udostępniającą metody do tworzenia, pobierania, aktualizowania i usuwania rekordów pracowników.

Klient jest budowany wizualnie za pomocą wizualnego edytora Java przy użyciu komponentów Swing. Wizualny edytor Java udostępnia zestaw klas pomocniczych (źródła danych, obiekty danych i konsolidatory) do łączenia się i pracy z usługą Web Service. Usługa Web Service jest wdrażana lokalnie w posiadanej instalacji serwera IBM WebSphere Application Server 6.0, a narzędzia pomagają wygenerować serwer proxy Java dla klienta na podstawie pliku WSDL (Web Services Description Language).

Patrz ukończony produkt

Cele edukacyjne

Ten kurs obejmuje następujące lekcje:

- Sposób używania wizualnego edytora Java do projektowania i określania układu interfejsu użytkownika
- Sposób wiązania elementów interfejsu z obiektami danych i usługą Web Service

Czas trwania

2 godziny i 15 minut

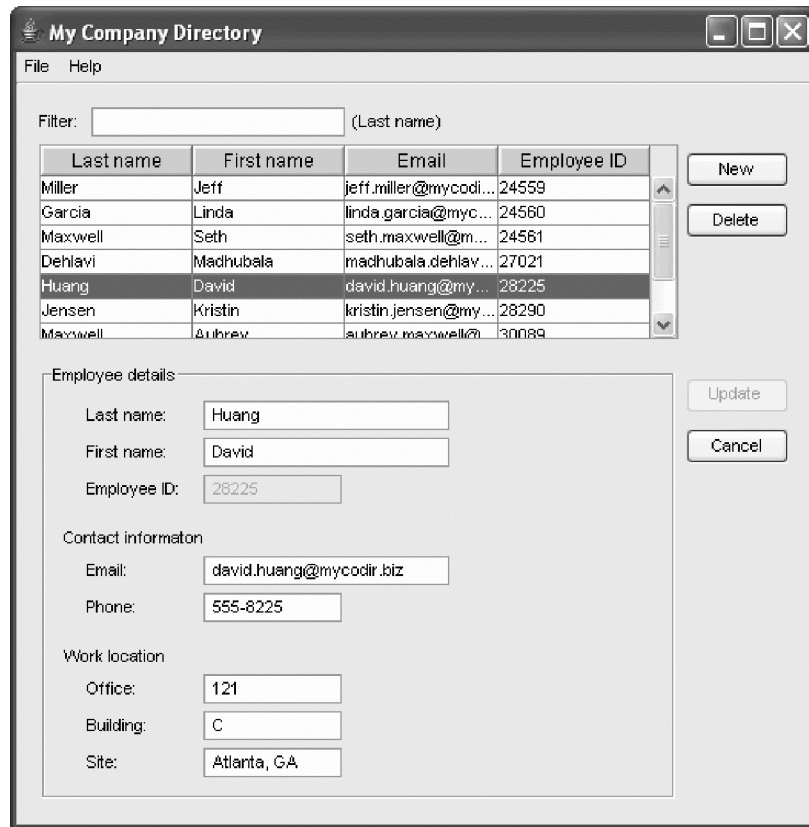
Informacje pokrewne

Wyświetl wersję w formacie PDF

Kurs: Witaj świecie Java

Wprowadzenie: Budowanie pełnego klienta Java korzystającego z usługi Web Service

Graficzny interfejs użytkownika (GUI) klienta został w większości wstępnie zbudowany pod względem wizualnym przy użyciu komponentów Swing. W pierwszym module użytkownik zakończy układanie istotnych komponentów interfejsu GUI przy użyciu wizualnego edytora Java. W drugim module użytkownik powiąże komponenty interfejsu GUI ze źródłem danych, usługami i obiektami usługi Web Service zwracanymi przez źródło danych. Podczas budowania aplikacji w wizualnym edytorze Java użytkownik będzie korzystał ze źródeł danych, obiektów danych i konsolidatorów, które są instancjami klas pomocniczych wygenerowanymi przez wizualny edytor Java i używanymi przez aplikację użytkownika.



Patrz obraz ukończonego produktu.:

Cele edukacyjne

Ten kurs obejmuje następujące lekcje:

- Sposób używania wizualnego edytora Java do projektowania i określania układu interfejsu użytkownika
- Sposób wiązania elementów interfejsu z obiektami danych i usługą Web Service

Czas trwania

Na ukończenie całego kursu użytkownik będzie potrzebował około 2 godziny i 30 minut.

Wymagania systemowe

- Serwer WebSphere Application Server w wersji 6.1. Ten serwer może być już zainstalowany razem z używanym produktem lub można użyć własnej autonomicznej instalacji. W scenariuszu tego kursu użytkownik otrzymuje polecenie wdrożenia przykładowej usługi Web Service na działającym lokalnie serwerze WebSphere Application Server.

Przykładowa usługa Web Service może działać na innych serwerach, ale wykonywane w tym kursie kroki przetestowano tylko na serwerze WebSphere Application Server w wersji 6.0 i 6.1.

Wymagania wstępne

Użytkownik powinien mieć wiedzę z następujących zagadnień:

- Podstawy programowania w języku Java
- Podstawowe zasady działania usługi Web Service
- Podstawowe umiejętności pracy w środowisku roboczym, takie jak praca z projektami oraz nawigowanie między perspektywami i widokami

Moduł 1: Projektowanie interfejsu GUI klienta w edytorze wizualnym

W tym module użytkownik nauczy się korzystać z wizualnego edytora Java w celu dodawania komponentów wizualnych do aplikacji, a następnie wizualnego ich rozmieszczania i ustawiania ograniczeń dotyczących położenia. W ostatniej lekcji w tym module opisano metodę uruchamiania plików Java w celu sprawdzenia wyglądu rzeczywistej aplikacji.

Zapamiętaj: Przed rozpoczęciem tego modułu użytkownik musi posiadać wstępnie wymaganą wiedzę opisaną we wprowadzeniu do kursu.

Cele edukacyjne

Po zakończeniu lekcji w tym module użytkownik będzie rozumiał i znał sposób wykonania następujących czynności:

- Dodawanie i układanie obiektu JTable w interfejsie Java
- Uruchamianie klasy wizualnej w celu sprawdzenia wyników pracy.

Czas trwania

Wykonanie poleceń zawartych w tym module zajmie około 15 minut.

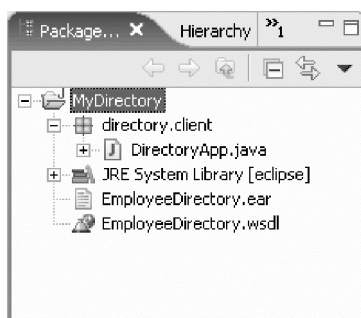
Lekcja 1.1: Konfigurowanie projektu Java

Podczas tej lekcji użytkownik skonfiguruje projekt MyDirectory, importując go do swojego obszaru roboczego. W projekcie tym występuje pojedyncza klasa Java, a także inne pliki, które będą używane później.

Ponieważ głównym celem tego kursu jest powiązanie komponentów wizualnych z usługą Web Service, większa część interfejsu GUI Java dla aplikacji "My Company Directory" została już wstępnie zaprojektowana.

Projekt MyDirectory jest głównym projektem Java, z którym użytkownik będzie pracował w ramach tego kursu. Zawiera on plik DirectoryApp.java, który jest plikiem Java zawierającym główną budowaną przez użytkownika aplikację Java. Dla ułatwienia w kursie dostępnych jest kilka wersji projektu MyDirectory: po jednej początkowej dla poszczególnych modułów i ostateczna wersja ukończonego projektu.

1. Zainportuj projekt MyDirectory.
2. W eksploratorze pakietów perspektywy Java projekt MyDirectory powinien wyglądać tak, jak na następującej ilustracji:



Punkt kontrolny lekcji

Podczas tej lekcji użytkownik zainportował przykładowy projekt MyDirectory, rozpoczynając w ten sposób niniejszy kurs.

W projekcie MyDirectory użyto następujących zasobów:

- DirectoryApp.java: Plik Java zawierający aplikację projektowaną w ramach tego kursu. Plik DirectoryApp.java znajduje się w pakiecie Java o nazwie directory.client.
- EmployeeDirectory.ear: Aplikacja korporacyjna zawierająca przykładową usługę Web Service. W module 2 użytkownik wdroży tę usługę Web Service w lokalnej instalacji serwera WebSphere Application Server 6.0.

- EmployeeDirectory.wsdl: Plik XML używający języka WSDL (Web Services Description Language) do opisania przykładowej usługi Web Service przygotowanej do wdrożenia przez użytkownika. W module 2 użytkownik użyje tego pliku WSDL do wygenerowania serwera proxy Java do użycia przez aplikację.

Lekcja 1.2: Dodanie tabeli pracowników i określenie jej układu

W tej lekcji użytkownik posłuży się wizualnym edytorem Java, aby dodać do aplikacji obiekty JScrollPane i JTable. W dalszych ćwiczeniach użytkownik zaprogramuje obiekt JTable do pobierania danych z usługi Web Service, która zwraca listę wszystkich pracowników w katalogu firmy.

Po dodaniu obiektu JTable użytkownik użyje widoku projektowania w edytorze wizualnym Java do dostosowania układu obiektu JTable, aby zapewnić zgodność z następującymi specyfikacjami:

- Rozciągnięcie obiektu JTable na trzy komórki w poziomie i dwie komórki w pionie
- Dodanie z lewej strony odstępów o szerokości 15 pikseli
- Zmiana nazwy obiektu JTable na employeesTable.

Pokaż

Otwarcie pliku DirectoryApp.java w wizualnym edytorze Java

Aby otworzyć plik DirectoryApp.java w wizualnym edytorze Java:

1. W widoku eksploratora pakietów perspektywy Java rozwiń projekt MyDirectory oraz pakiet directory.client.
2. Kliknij prawym przyciskiem myszy plik DirectoryApp.java i wybierz opcję **Otwórz za pomocą** → **Edytor graficzny**. Wizualny edytor Java załaduje klasę Java i wyświetli projekt w obszarze kanwy graficznej.

Wskazówka:

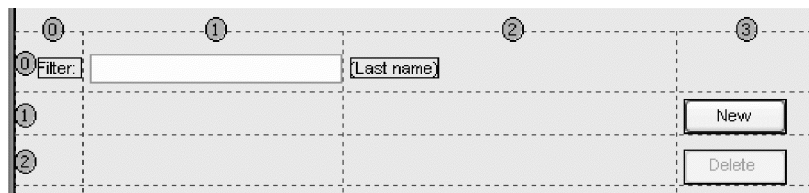
- Aby zmienić wygląd i zachowanie używane przez wizualny edytor Java, wybierz opcję **Okno** → **Preferencje** → **Java** → **Edytor graficzny**, a następnie określ wygląd i zachowanie Swing. Ustawione preferencje zostaną aktywowane przy następnym otwarciu klasy. W tym kursie jest używany wygląd i zachowanie systemu Windows.
- Aby ustawić edytor wizualny jako domyślny edytor wszystkich plików Java, kliknij opcję **Okno** → **Preferencje** i przejdź do strony **Środowisko robocze** → **Powiązania plików**, aby zdefiniować własne preferencje.

Dodanie obiektu JTable do obiektu JScrollPane

Główne okno w pliku DirectoryApp.java używa obiektu JFrame z obiektem JPanel jako głównego panelu zawartości. Obiekt JPanel w tej aplikacji nosi nazwę jContentPane. Obiekt jContentPane został ustawiony do korzystania z typu menedżera układu o nazwie GridBagLayout. Typ GridBagLayout jest wszechstronnym schematem układu opartym na siatce komórek, które mogą być zajmowane przez elementy wizualne. Wizualny edytor Java ułatwia pracę z typem GridBagLayout, wyświetlając ramki siatki. Wyświetla on również znaczniki rozmieszczenia podczas upuszczania nowych komponentów na siatkę oraz wyświetla uchwyty komponentów, których wielkość jest zmieniana lub które są przenoszone na siatkę GridBagLayout.

Aby dodać tabelę pracowników (javax.swing.JTable) do interfejsu użytkownika w pliku DirectoryApp.java:

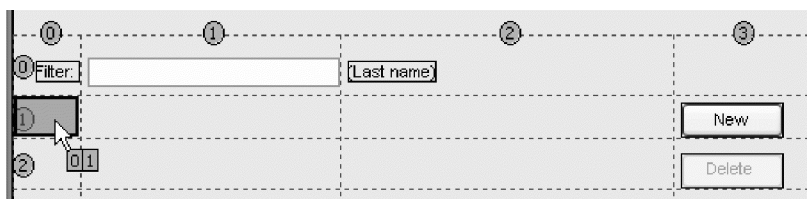
1. Kliknij prawym przyciskiem myszy obiekt jContentPane w widoku projektu lub widoku komponentów bean Java i wybierz opcję **Wyświetl siatkę**. Czerwona kropkowana linia oznacza granicę siatki, a niebieskie kółka z numerami wskazują numery wierszy i kolumn. Łatwo na przykład zauważyć, że przycisk **Nowy** zajmuje komórkę w wierszu 1 (współrzędna y siatki) i kolumnie 3 (współrzędna x siatki).



2. Z palety wizualnego edytora Java wybierz komponent Swing **JTable on JScrollPane** , który znajduje się w szufladzie **Komponenty Swing** palety.

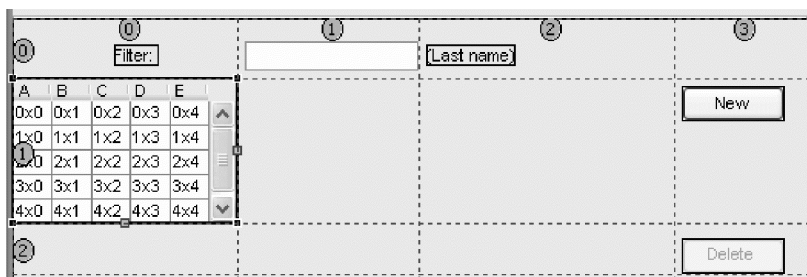
Wskazówka: Domyślnie paleta jest zwinięta z prawej strony obszaru projektowania. Paletę można przenosić, a także zmieniać jej wielkość.

3. Ustaw wskaźnik myszy nad komórką siatki w kolumnie 0 i wierszu 1:



- W miarę przesuwania wskaźnika myszy nad siatką wskazuje on dwa numerowane kwadraty określające współrzędne x i y siatki zależne od położenia wskaźnika.
- W przypadku ustawienia wskaźnika myszy bezpośrednio na granicy siatki można utworzyć nowe wiersze i kolumny, co spowoduje zmianę numeracji istniejących już wierszy i kolumn. W takim przypadku żółte kwadraty przy wskaźniku myszy, żółte paski między siatkami i żółte etykiety kolumny i wiersza oznaczają taką sytuację i określają wpływ takiego umieszczenia elementu.

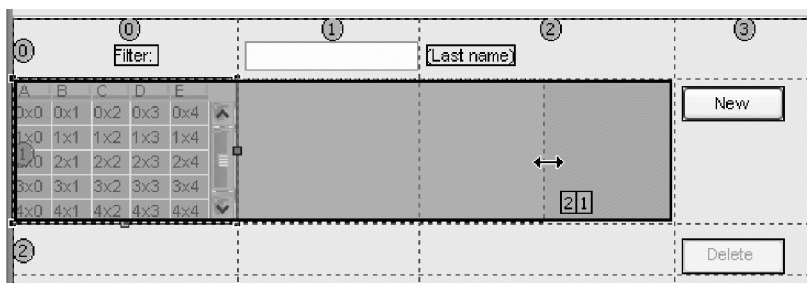
4. Kliknij lewym przyciskiem myszy, aby upuścić obiekty JScrollPane i JTable w komórce w kolumnie 0 i wierszu 1:



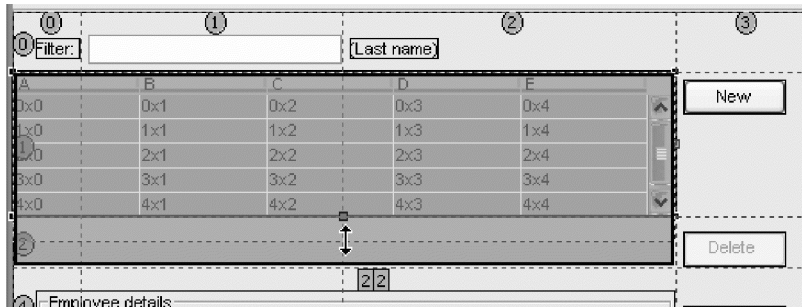
Rozciąganie obiektów JScrollPane i JTable na kilka kolumn i wierszy siatki

Teraz należy rozciągnąć obiekt JScrollPane (i jego element potomny JTable), aby zajmował trzy kolumny i dwa wiersze, co ułatwi tworzenie odstępów i zmianę wielkości. Aby rozszerzyć tabelę na wiele kolumn i wierszy:

1. Wybierz obiekt JScrollPane w obszarze projektowania lub w widoku komponentów bean Java (powinien nadal być jeszcze wybrany, ponieważ dopiero został dodany). Zwróć uwagę na małe zielone kwadraty z prawej strony i u dołu obiektu JScrollPane. Te uchwyty zmiany wielkości służą do przeciągnięcia obiektu JScrollPane tak, aby zajmował wiele kolumn i wierszy.
2. Kliknij i przytrzymaj lewy przycisk myszy na zielonym uchwycie z prawej strony obiektu JScrollPane.
3. Przeciągnij wskaźnik myszy w prawo do chwili, gdy współrzędne rozmieszczenia wskażą kolumnę 2 i wiersz 1. Również ciemnoszary cień będzie wskazywał komórki, które zostaną zajęte przez komponent po zwolnieniu przycisku myszy.



4. Zwolnij przycisk myszy. Obiekt JScrollPane zajmuje teraz trzy kolumny.
5. Wykonaj analogiczną czynność, aby przeciągnąć dolny uchwyt obiektu JScrollPane aż osiągnie on wiersz 2:



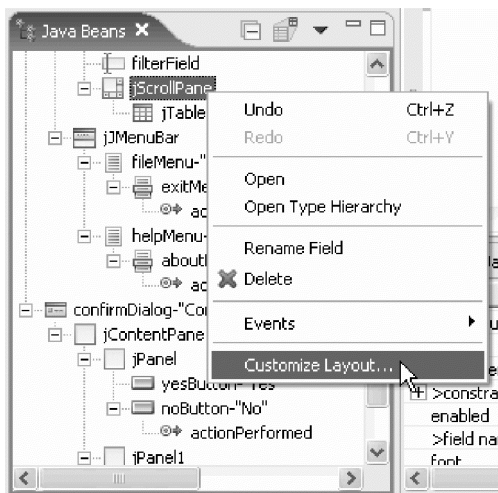
Dostosowywanie odstępów obiektu JScrollPane w ramach siatki GridBag

Kolejną funkcją menedżera GridBagLayout jest możliwość określenia różnych ograniczeń umożliwiających dalsze dostosowanie układu. Na przykład można określić następujące ograniczenia:

- **zakotwiczenie:** Komponentowi można nadać orientację zakotwiczenia w ramach komórki, co wpływa na sposób przemieszczania komponentu w przypadku zmiany wielkości aplikacji przez użytkownika. Na przykład komponent może zostać zakotwiczony u góry z lewej strony, na środku z lewej strony, w centrum lub u dołu z prawej strony.
- **wypełnienie:** Komponent może zajmować cały dostępny obszar w ramach komórki lub komórek zarówno w poziomie, w pionie, jak i w obu kierunkach.
- **odstęp:** Komponent może mieć własne dopełnienie u góry, u dołu, z lewej i prawej strony w celu zapewnienia odstępu między nim a brzegiem siatki.

Aby dostosować zakotwiczenie, wypełnienie i odstępy obiektu JScrollPane:

1. Kliknij prawym przyciskiem myszy obiekt JScrollPane w widoku projektu lub widoku komponentów bean Java i wybierz opcję **Dostosuj układ**.



Wskazówka: Okno dialogowe Dostosowanie układu może pozostawać otwarte podczas wybierania i zmieniania układu różnych komponentów. Okno Dostosowanie układu można otworzyć w dowolnej chwili, klikając przycisk Dostosuj układ na pasku menu:



2. Na karcie Składnik w oknie dialogowym Dostosowanie układu sprawdź, czy jest naciśnięty przycisk zakotwiczenia w centrum.
3. Upewnij się, że są naciśnięte oba przyciski **wypełnienia poziomego** i **wypełnienia pionowego**.

4. Dodaj z lewej strony odstęp o wielkości 15 (piksli), aby zapewnić miejsce z lewej strony obiektu JScrollPane analogicznie do innych elementów wizualnych aplikacji.

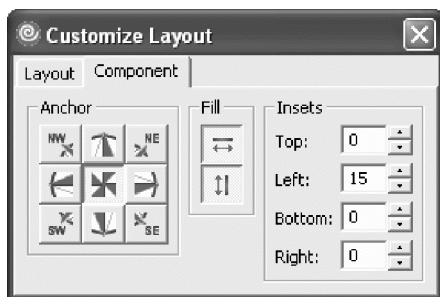
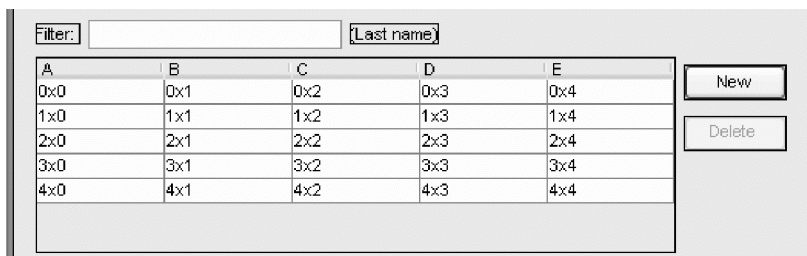


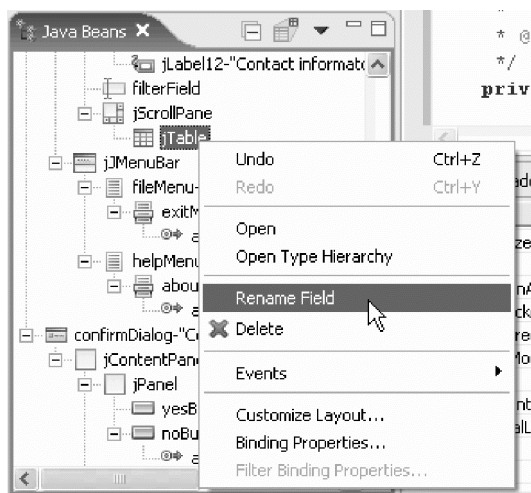
Tabela jest teraz wyrównana na przykład z etykietą **Filtr**.



Zmianianie nazwy nowego obiektu JTable na odpowiednią wartość i ustawianie go na wybór pojedynczego wiersza

Ponieważ później tabela będzie jeszcze używana, warto zmienić nazwę instancji obiektu JTable i jego metody pobierającej. Aby zmienić nazwę tabeli:

1. W widoku komponentów bean Java kliknij prawym przyciskiem myszy komponent jTable i z menu podręcznego wybierz opcję **Zmień nazwę pola**.



2. Wpisz `employeesTable` i kliknij przycisk **OK**. Obiekt JTable ma teraz nazwę `employeesTable`, a metoda tworzenia jego instancji nosi nazwę `getEmployeesTable`.
3. Ustaw tabelę tak, aby można było wybrać tylko jeden wiersz:
 - a. Wybierz tabelę `employeesTable` w widoku projektowania.
 - b. W widoku Właściwości wybierz właściwość **selectionMode** i ustaw jej wartość na `SINGLE_SELECTION`.

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. Zapisz plik DirectoryApp.java.

Punkt kontrolny lekcji

W tej lekcji użytkownik poznał sposób dodawania tabeli do istniejącego interfejsu użytkownika za pomocą edytora wizualnego. Następnie użytkownik dowiedział się, jak dostosować jej układ, położenie i odstępy.

Lekcja 1.3: Uruchomienie klasy wizualnej

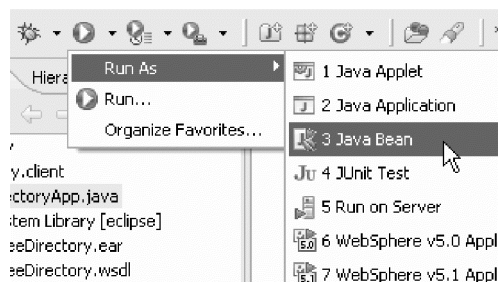
Aplikację Java można już uruchomić, aby sprawdzić jej wygląd. Środowisko robocze i edytor wizualny ułatwiają szybkie uruchomienie aplikacji, przy czym kroki te można powtórzyć w dowolnej chwili podczas projektowania, aby sprawdzić bieżący wygląd i zachowanie klasy w czasie wykonywania.

Pokaż

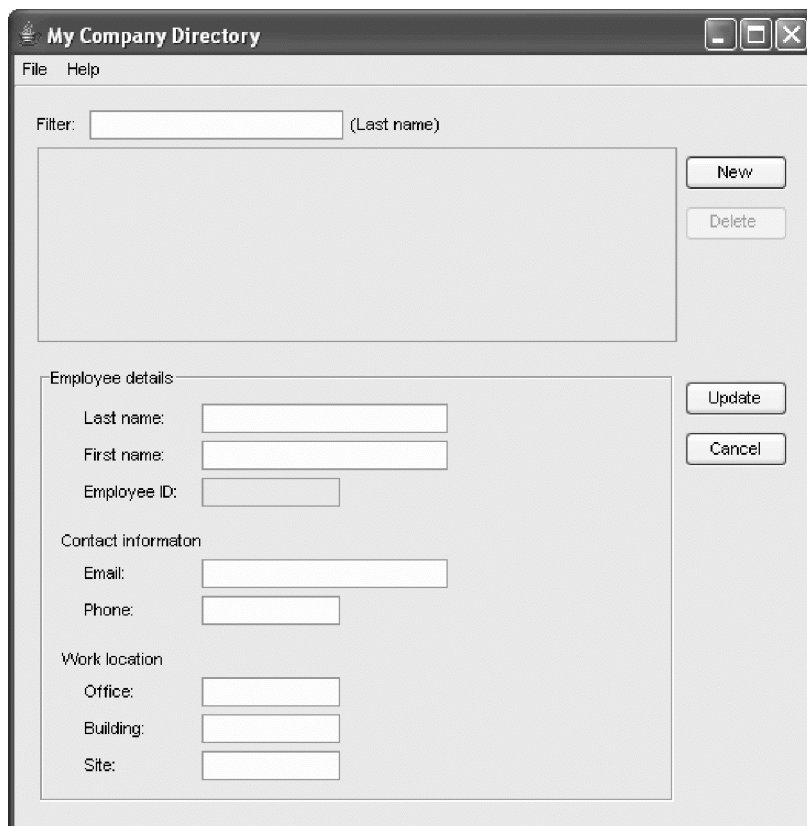
Wizualny edytor Java udostępnia program uruchamiający Java Bean umożliwiający uruchamianie klas bez metody main(). Podczas uruchamiania klasy wizualnej aplikacja działa na oddzielnej maszynie wirtualnej (VM). W przypadku uruchomienia klasy wizualnej jako aplikacji Java program uruchamiający próbuje wykonać metodę main() w klasie. Na potrzeby tego kursu aplikacja zawiera metodę main(), która wywołuje i wyświetla obiekt JFrame pliku DirectoryApp, dzięki czemu można ją uruchomić jako aplikację lub komponent bean Java.

Aby uruchomić plik DirectoryApp.java jako komponent bean Java:

1. Upewnij się, że plik DirectoryApp.java jest otwarty w edytorze wizualnym Java.
2. Na pasku menu kliknij opcję **Uruchom** → **Uruchom jako** → **Komponent Java Bean**.



Wskazówka: Aplikacja zostanie otwarta na pulpicie przy użyciu wyglądu i zachowania Swing zdefiniowanego w preferencjach edytora graficznego (**Okno** → **Preferencje** → **Java** → **Edytor graficzny**). Ewentualnie można kliknąć opcję **Uruchom** → **Uruchom** i zdefiniować wygląd i zachowanie dla konkretnej konfiguracji uruchamiania na potrzeby uruchamiania tego komponentu bean Java. Jeśli aplikacja zostanie uruchomiona jako aplikacja, a nie komponent bean, będzie ona używała również wyglądu i zachowania systemu Windows, ponieważ zostało to zdefiniowane w metodzie main(). Na zrzutach ekranu używanych w tym kursie przedstawiony jest wygląd i zachowanie systemu Windows.



Punkt kontrolny lekcji

Ponieważ na razie został zaprojektowany tylko interfejs i nie zostały zaprogramowane żadne połączenia danych ani funkcje zdarzeń, aplikacji brak funkcjonalności. Można jednak zobaczyć podstawowy układ i wygląd tak, jak zobaczy to użytkownik. Można spróbować kliknąć niektóre przyciski, ale łatwo zauważyć, że nie wykonują one żadnych funkcji. Zaimplementowane zostały już za to menu Plik i Pomoc. Można spróbować ich użyć, aby zobaczyć co robią, a także można przejrzeć kod Java, aby zobaczyć, jak zostały one zaimplementowane za pomocą zdarzeń `ActionPerformed`.

Zgromadzone doświadczenie

Ten moduł wprowadził użytkownika w zagadnienia projektowania interfejsu pełnego klienta za pomocą wizualnego edytora Java. Jednak oprócz zaprojektowania wyglądu klienta istnieje jeszcze dużo więcej czynności do wykonania w celu zapewnienia jego użyteczności. Zazwyczaj należy uwzględnić logikę zachowania zdarzeń i pozostałą logikę aplikacji, a także, w tym przypadku, powiązanie elementów wizualnych z pewnego rodzaju źródłem danych.

W tym module użytkownik dowiedział się, jak wykonać następujące zadania:

- Importowanie projektu Java za pomocą importu typu Project Interchange
- Dodawanie obiektu `JTable` z obiektem `JScrollPane` do klasy wizualnej
- Używanie menedżera `GridBagLayout` do wizualnego ułożenia tabeli w pełnym kliencie
- Uruchamianie aplikacji w celu sprawdzenia rzeczywistego wyglądu pełnego klienta Java

W następnym module (Moduł 2: Wiązanie komponentów wizualnych z usługą Web Service) użytkownik przekształci prosty interfejs aplikacji Firmowa książka adresowa w rozbudowanego pełnego klienta z dostępem do metod usługi Web Service służących do tworzenia, pobierania, aktualizowania i usuwania rekordów pracowników z katalogu firmy.

Moduł 2: Wiązanie elementów wizualnych z usługą Web Service

W tym module przedstawiono sposób wiązania elementów wizualnych aplikacji Firmowa książka adresowa (przycisków, tabeli pracowników, pól i innych czynności) z usługą Web Service. Usługa Web Service udostępnia rzeczywiste funkcje tworzenia, pobierania, aktualizowania i usuwania pracowników z przykładowego katalogu.

Cele edukacyjne

Po zakończeniu lekcji w tym module użytkownik będzie rozumiał i znał sposób wykonania następujących czynności:

- Wiązanie tabeli ze źródłem danych usługi Web Service
- Wiązanie pól z obiektami
- Programowanie czynności w przyciskach

Wykonanie tego modułu zajmuje około **2 godziny**.

Lekcja 2.1: Zainstalowanie i wdrożenie usługi Web Service

W tym ćwiczeniu użytkownik zainstaluje plik przykładowej aplikacji korporacyjnej (EAR) na serwerze WebSphere Application Server w wersji 6.1 i wdroży usługę Web Service o nazwie EmployeeDirectory. Aplikacja będzie korzystać z tej usługi Web Service do tworzenia, odczytywania, aktualizowania i usuwania rekordów pracowników.

Przed rozpoczęciem należy wykonać *jeden z poniższych punktów*, aby zapewnić odpowiedni stan początkowy projektu MyDirectory:

- Wykonaj “Moduł 1: Projektowanie interfejsu GUI klienta w edytorze wizualnym” na stronie 3.
lub
- Zaimportuj projekt MyDirectory przygotowany dla modułu 2

Wskazówka: Jeśli podczas importowania nie zostanie określona inna nazwa projektu, treść projektu MyDirectory użytkownika zostanie zastąpiona.

Projekt Java MyDirectory obejmuje plik EmployeeDirectory.ear. Za pomocą konsoli administracyjnej produktu WebSphere należy zainstalować aplikację korporacyjną EmployeeDirectory znajdującą się w pliku EAR. Podczas instalowania aplikacji zostanie również wdrożona usługa Web Service zawarta w aplikacji. Tej wdrożonej usługi Web Service używa ukończona aplikacja Firmowa książka adresowa.

Aby zainstalować przykładową aplikację EmployeeDirectory i wdrożyć usługę Web Service w środowisku serwera WebSphere Application Server w wersji 6.1:

1. Uruchom w środowisku roboczym instancję swojego serwera aplikacji. Istnieje kilka różnych sposobów uruchomienia serwera, ale poniższe kroki opisują sposób wykonania tej czynności z poziomu środowiska roboczego:
 - a. Otwórz widok Serwery. Aby dodać widok Serwery do perspektywy Java, kliknij opcję **Okno → Pokaż widok → Inne → Serwery**.
 - b. W widoku Serwery wyświetlana jest lista zainstalowanych i skonfigurowanych serwerów.
 - c. Kliknij prawym przyciskiem myszy swój serwer i wybierz opcję **Uruchom**. Gdy w widoku Serwery status serwera zostanie wyświetlony jako **Uruchomiony** lub w konsoli zostanie wyświetlony komunikat **Server server1 open for e-business**, serwer został pomyślnie uruchomiony. Można teraz uruchomić konsolę administracyjną.

Uwaga: Jeśli w widoku Serwery nie ma żadnej instancji serwera, utwórz nowy serwer:

- a. Kliknij prawym przyciskiem myszy w widoku Serwery i wybierz opcję **Nowy → Serwer**.
- b. Użyj kreatora Nowy serwer, aby dodać serwer WebSphere Application Server w wersji 6.1.

2. Uruchom konsolę administracyjną WebSphere. Również w tym przypadku istnieją inne sposoby uruchamiania konsoli administracyjnej, ale poniższe instrukcje opisują sposób wykonania tej czynności z poziomu środowiska roboczego:
 - a. W widoku Serwery kliknij prawym przyciskiem myszy właśnie uruchomiony serwer i wybierz opcję **Uruchom konsolę administracyjną**. Konsola administracyjna WebSphere zostanie otwarta w oknie przeglądarki.
 - b. Podaj identyfikator użytkownika i kliknij przycisk **Zaloguj**. Zostanie otwarta strona powitania konsoli administracyjnej. Podany identyfikator użytkownika służy tylko do śledzenia zmian wprowadzanych przez użytkownika w danych konfiguracyjnych serwera.
3. Za pomocą konsoli administracyjnej zainstaluj aplikację korporacyjną EmployeeDirectory.ear znajdującą się w projekcie MyDirectory. Konsola administracyjna korzysta z kreatora w celu zapewnienia pomocy podczas instalowania aplikacji, w którym do chwili ustawienia wszystkich opcji należy klikać przycisk **Dalej**, aby przechodzić do kolejnych stron. Aby zainstalować przykładową aplikację korporacyjną zawierającą usługę Web Service dla tego kursu:
 - a. Z lewej strony konsoli administracyjnej rozwiń opcję **menu Aplikacje**, a następnie kliknij opcję **Zainstaluj nową aplikację**.
 - b. Wybierz opcję **Lokalny system plików** i w polu **Określ ścieżkę** wprowadź pełną ścieżkę do pliku EmployeeDirectory.ear znajdującego się w projekcie MyDirectory. Wskazówka: Aby uzyskać pełną ścieżkę, kliknij prawym przyciskiem myszy plik EmployeeDirectory.ear w eksploratorze pakietów i wybierz opcję **Właściwości**. Na stronie Właściwości wyświetlane jest położenie pliku, które można skopiować i wkleić w polu **Określ ścieżkę**.
 - c. Klikaj przycisk **Dalej**, dopóki nie zostanie wyświetlona strona **Wybór opcji instalacji**.
 - d. Wybierz opcję **Przeprowadź wdrożenie usług Web Services**.
 - e. Klikaj przycisk **Dalej**, dopóki nie zostanie wyświetlona strona **Podsumowanie**, a następnie kliknij przycisk **Zakończ**.
 - f. Kliknij odsyłacz **Zapisz w konfiguracji głównej** po wyświetleniu monitu o zastosowanie wprowadzonych zmian do konfiguracji lokalnej. Przejrzyj zmiany i kliknij przycisk **Zapisz**.
4. Za pomocą konsoli administracyjnej uruchom aplikację EmployeeDirectory:
 - a. Kliknij opcję **Aplikacje** → **Aplikacje korporacyjne**. Aplikacja EmployeeDirectory jest wyświetlona jako aplikacja zainstalowana na serwerze, ale jej status to Zatrzymany.

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

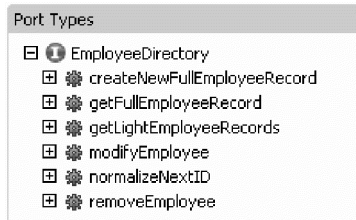
- b. Zaznacz pole wyboru obok aplikacji EmployeeDirectory i kliknij przycisk **Uruchom**. Zostanie wyświetlony komunikat z informacją, że aplikacja EmployeeDirectory została pomyślnie uruchomiona, a jej ikona statusu zmieni się w zieloną strzałkę.

Aplikacja EmployeeDirectory działa teraz na porcie 9080 hosta lokalnego, umożliwiając dostęp do usługi Web Service. Po zakończeniu tego kursu można wrócić do konsoli administracyjnej, zatrzymać aplikację EmployeeDirectory, a następnie ją zdeinstalować.

Po otwarciu pliku EmployeeDirectory.wsdl znajdującego się w projekcie MyDirectory (powinien on zostać domyślnie otwarty w graficznym edytorze WSDL) można zapoznać się z właśnie wdrożoną usługą Web Service. Jeśli plik WSDL

nie zostanie otwarty w edytorze WSDL, w środowisku roboczym mogły nie zostać włączone funkcje produktu Web Service Developer. Możliwości środowiska roboczego można określić w preferencjach (**Okno** → **Preferencje** → **Środowisko robocze** → **Możliwości**).

Poniższy obraz z edytora WSDL przedstawia operacje dostępne w usłudze EmployeeDirectory:



W edytorze WSDL można sprawdzić każdą operację oraz odpowiadające jej komunikaty żądań i komunikaty zwrotne. Może to pomóc w zrozumieniu działania usługi Web Service oraz sposobu jej używania w kolejnych ćwiczeniach.

Lekcja 2.2: Powiązanie tabeli pracowników ze źródłem danych usługi Web Service

Aplikacja Firmowa książka adresowa wyświetla listę wszystkich aktualnych rekordów pracowników w katalogu. Rekordy są wyświetlane w obiekcie JTable (employeesTable) w kolumnach z możliwością sortowania, które zawierają nazwisko, imię, adres e-mail i identyfikator pracownika. W celu pobrania rekordów do tabeli należy powiązać obiekt employeesTable z obiektem danych zwracanym przez źródło danych przykładowej usługi Web Service.

Pokaż

Przegląd obiektów danych, źródeł danych i konsolidatorów

Aby pobrać lokalny obiekt danych do pracy z obiektem employeesTable, należy użyć edytora wizualnego w celu dodania do aplikacji źródła danych. Źródło danych łączy się z serwerem proxy przykładowej usługi Web Service i wykrywa metody usługi dostępne dla aplikacji. Następnie należy wybrać metodę usługi getLightEmployeeRecord dostępną w źródle danych. Wreszcie należy powiązać obiekt employeesTable w aplikacji z polami zwracanymi w obiekcie danych wiersza (lightEmployeeRecordRows).

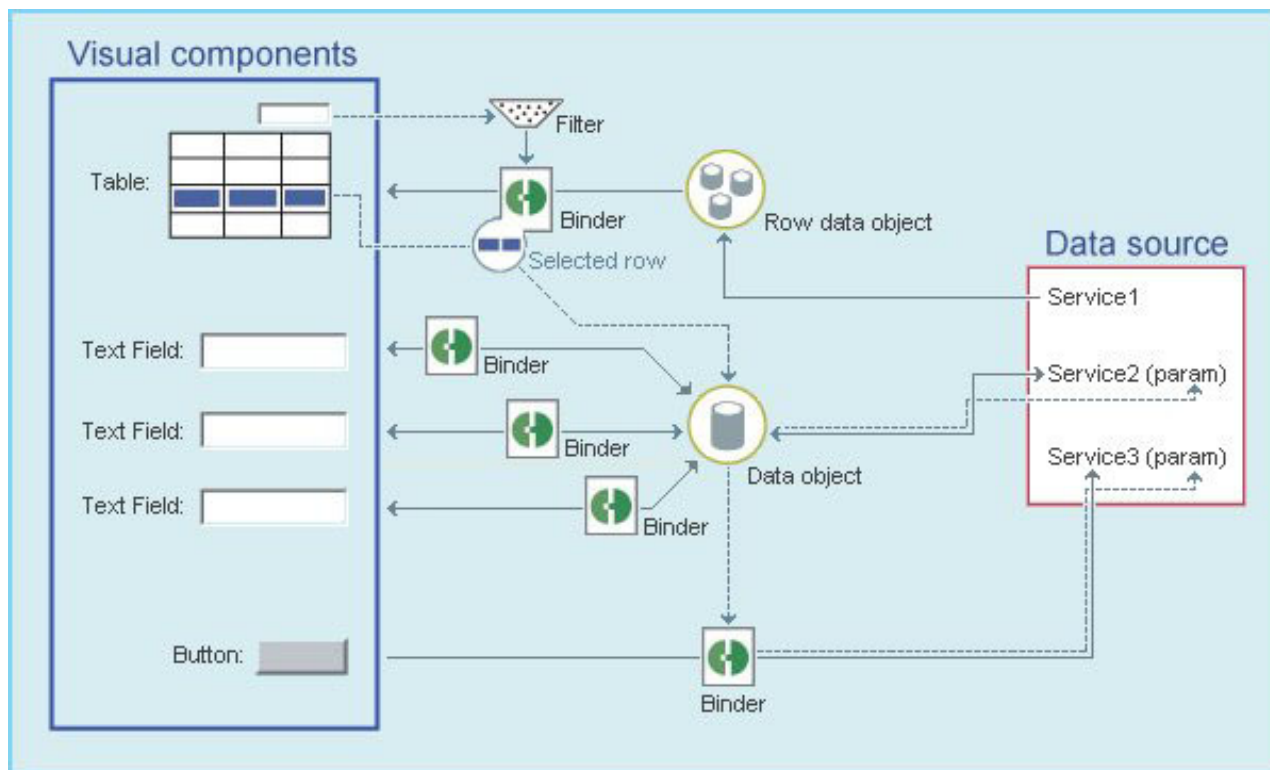
Wszystkie te źródła danych i obiekty danych można szybko i łatwo utworzyć za pomocą wbudowanych klas konsolidatora w wizualnym edytorze Java. Edytor wizualny udostępnia zestaw ogólnych interfejsów i klas generowanych w projekcie w miarę wiązania komponentów wizualnych z fabrykami danych. Klasy konsolidatora są domyślnie generowane w pakiecie o nazwie jve.generated. Edytor wizualny udostępnia klasy konsolidatora jako ogólną implementację, którą można następnie dostosować i rozszerzyć w celu spełnienia wymagań aplikacji. W ramach tego kursu zostanie przedstawiona wszechstronność i elastyczność nawet podstawowego i prostego zastosowania domyślnych klas konsolidatora.

Ważne: Przed rozpoczęciem tego ćwiczenia zdecydowanie zaleca się przeczytanie następujących tematów pomocy. Tematy te pozwolą uzyskać więcej informacji na temat funkcjonalności i logiki obiektów danych, źródeł danych i konsolidatorów udostępnianych przez wizualny edytor Java:

- Przegląd konsolidatorów danych
- Skorowidz funkcji API konsolidatora

W ramach tego kursu w aplikacji użyte zostanie źródło danych usługi Web Service, kilka typów obiektów danych i kilka typów konsolidatorów. Podczas dodawania instancji tych obiektów do aplikacji edytor wizualny dodaje niezbędne klasy do pakietu jve.generated projektu, gdzie można rozszerzyć, zastąpić lub przebudować logikę powiązania danych. Wizualny edytor Java zapewnia wizualną obsługę obiektów powiązań, wyświetlając w wolnym obszarze formularza w widoku projektowania obiekty danych, źródła danych i konsolidatory używane przez aplikację. Edytor wizualny rysuje linie między elementami wizualnymi i obiektami danych oraz źródłami danych w celu wskazania bieżących powiązań dla dowolnego wybranego obiektu.

Następujący diagram jest przykładowym przeglądem sposobu interakcji komponentów wizualnych, konsolidatorów, obiektów danych i źródeł danych. Aplikacja budowana w ramach tego kursu ilustruje trochę bardziej złożone i twórcze zastosowanie konsolidatorów. Ten diagram niedokładnie reprezentuje konsolidatory, obiekty danych i źródła danych w przykładowej budowanej aplikacji.



Rysunek 1. Na tym diagramie przedstawiono przykładową relację między komponentami wizualnymi, konsolidatorami, obiektami danych i źródłami danych

Na Rysunku 1 każdy komponent wizualny ma własny konsolidator łączący go z obiektem danych lub, w przypadku przycisku, źródłem danych. Konsolidatory pól tekstowych wiążą pole z konkretną właściwością obiektu danych. Zarówno obiekt danych wiersza, jak i obiekt danych na tym diagramie pobierają swoje dane z bezpośrednich wywołań usługi źródła danych. Obiekt danych dla pól tekstowych używa wartości klucza z wybranego wiersza w tabeli jako argumentu wywołania Usługi2, zwracającej pełny rekord, który najprawdopodobniej zawiera więcej informacji o wybranym wierszu w tabeli. Z kolei ten pełny rekord jest używany jako argument dla konsolidatora czynności przycisku podczas wywoływania Usługi3, która może być metodą aktualizującą wartości wprowadzone w polach. Więcej szczegółowych objaśnień dotyczących obiektów danych, konsolidatorów danych i źródeł danych można uzyskać, korzystając z wcześniej udostępnionych odsyłaczy.

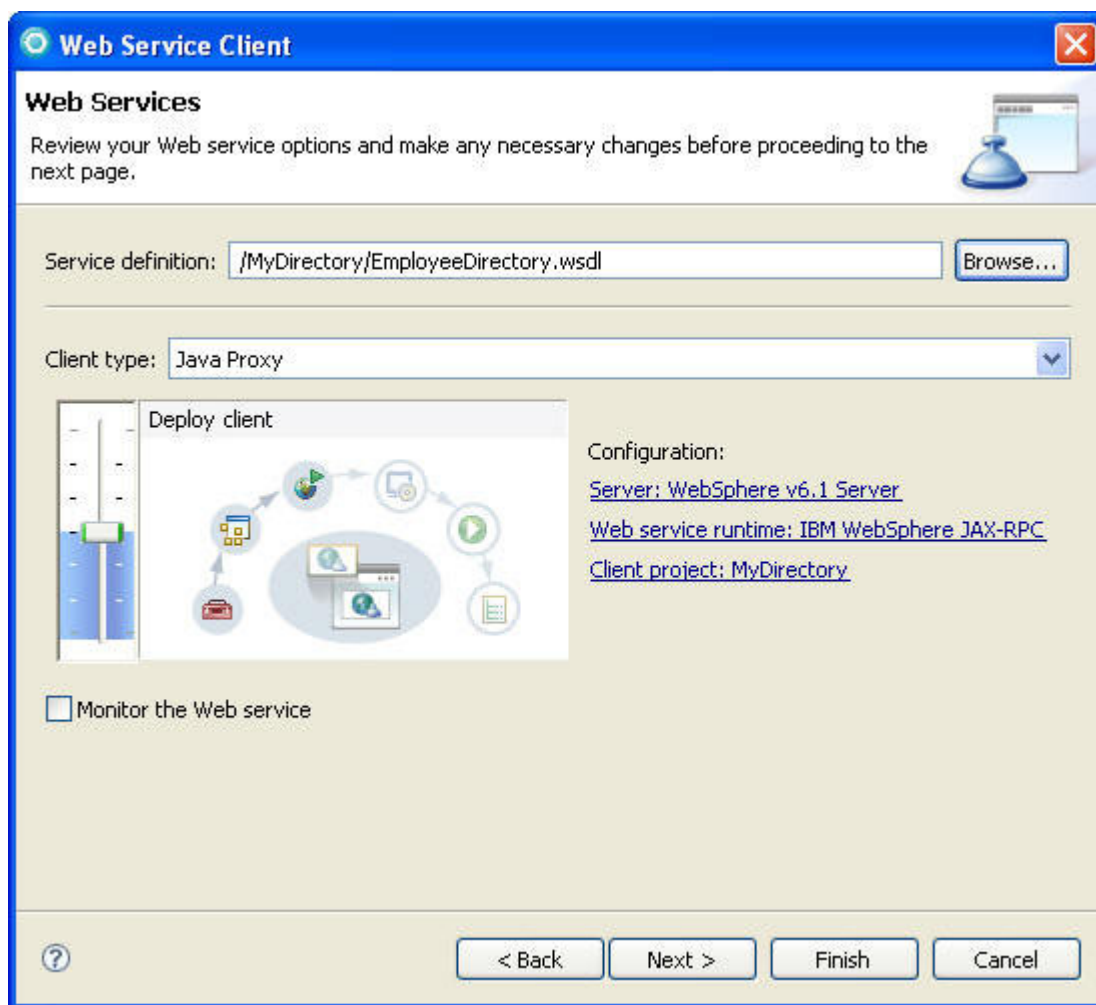
Generowanie w projekcie serwera proxy Java usługi Web Service za pomocą dostarczonego pliku WSDL

Aby móc korzystać z usługi Web Service działającej na serwerze, aplikacja Java wymaga serwera proxy Java, czyli klienta, do współpracy z usługą. Przy użyciu pliku WSDL można wygenerować w projekcie Java serwer proxy Java, korzystając z kreatora klienta usługi Web Service. Projekt MyDirectory zawiera plik EmployeeDirectory.wsdl, który zostanie użyty do wygenerowania serwera proxy. Po wygenerowaniu serwera proxy Java można utworzyć źródło danych reprezentujące usługę Web Service i rozpocząć wiązanie elementów wizualnych.

Ważne: W pliku WSDL używanym w tym ćwiczeniu przyjęto, że użytkownik wdrożył usługę Web Service w lokalnej instalacji serwera WebSphere Application Server i użył domyślnego portu dla hosta lokalnego (<http://localhost:9080>). W przypadku innego wdrożenia pliku EAR należy odpowiednio zmodyfikować plik WSDL przed kontynuowaniem pracy.

Aby wygenerować w projekcie serwer proxy Java usługi Web Service:

1. W menu głównym kliknij opcję **Plik** → **Nowy** → **Inne** i wybierz kreatora **Usługi Web Services** → **Klient Web Service**. Jeśli kategoria Usługi Web Services nie jest wyświetlana, wybierz opcję **Wyświetl wszystkie kreatory**.
2. Za pomocą kreatora zdefiniuj klienta usługi Web Service:
 - a. W polu **Definicja usługi** wprowadź nazwę pliku WSDL dostępnego w projekcie MyDirectory: `/MyDirectory/EmployeeDirectory.wsdl`.
 - b. W polu **Typ klienta** wybierz opcję **Serwer proxy Java**.
 - c. Pasek suwaka ustaw na opcję **Wdrażanie klienta**.
 - d. Upewnij się, że serwer i środowisko wykonawcze usługi Web Service są odpowiednio skonfigurowane dla używanego serwera. Opisane w tym kursie kroki zostały przetestowane na serwerach WebSphere 6.0 i WebSphere 6.1 ze środowiskiem wykonawczym IBM WebSphere JAX-RPC.
 - e. Upewnij się, że klient serwera proxy Java jest wyprowadzony do projektu MyDirectory.



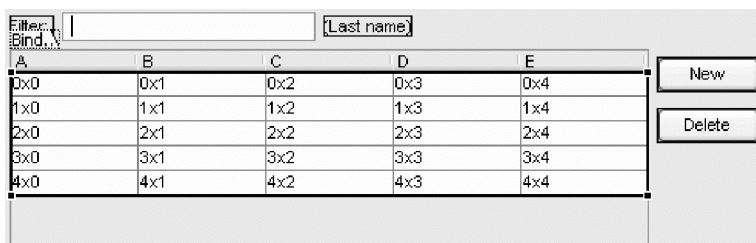
3. Kliknij przycisk **Zakończ**. Kreator klienta usługi Web Service generuje w projekcie użytkownika serwer proxy Java w nowym pakiecie (directory.service).

Powiązanie tabeli employeesTable z obiektem danych wiersza zwracanym przez usługę Web Service

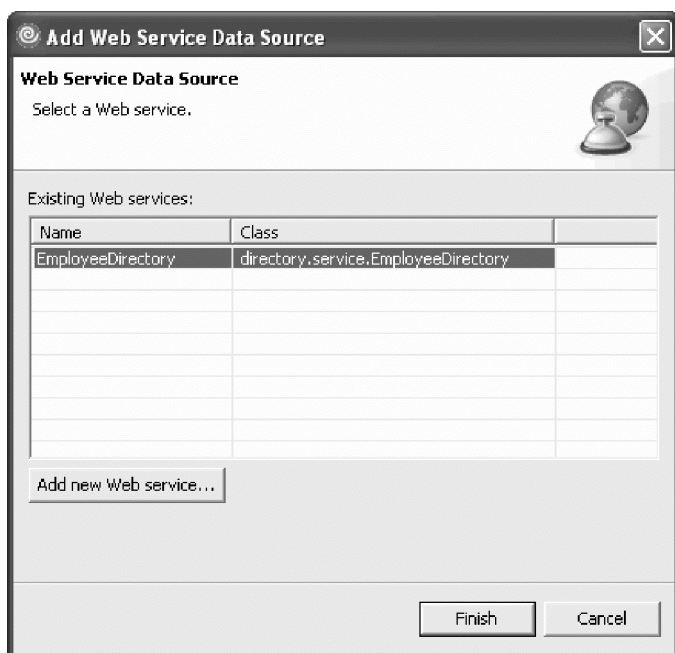
Ponieważ tabela employeesTable jest pierwszym komponentem wizualnym wiązaniem w tej aplikacji, należy utworzyć źródło danych wskazujące serwer proxy przykładowej usługi Web Service, który został właśnie dodany do projektu. Podczas wiązania pozostałych komponentów wizualnych w następnych ćwiczeniach użytkownik ponownie użyje tego źródła danych. W tym kroku użytkownik doda źródło danych usługi Web Service oraz obiekt danych lightEmployeeRecordRows.

Aby powiązać tabelę pracowników:

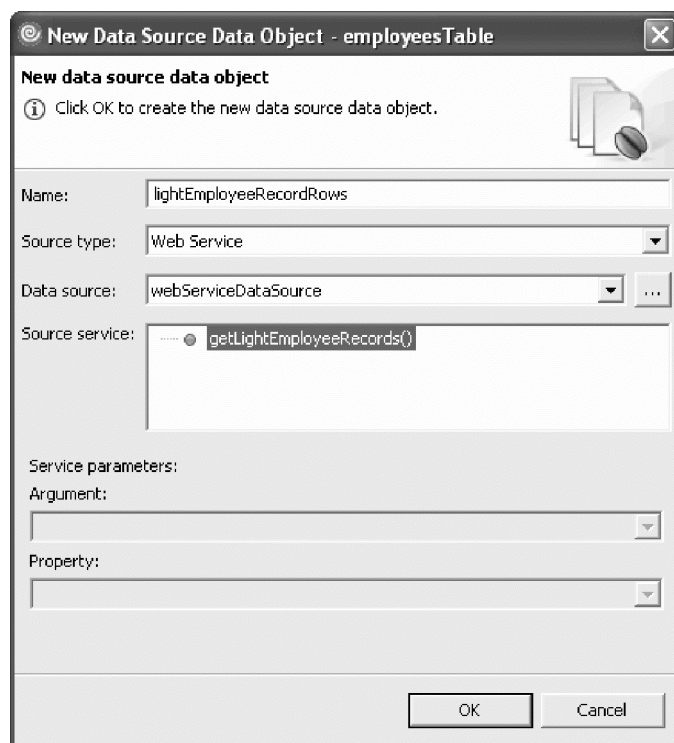
1. W widoku komponentów bean Java lub w widoku projektowania wybierz tabelę employeesTable (upewnij się, że nie został wybrany jej obiekt nadrzędny JScrollPane). W obszarze projektowania wyświetlona zostanie u góry tabeli employeesTable niewielka zakładka o nazwie **Wiązanie**.



2. Kliknij zakładkę **Wiązanie** tabeli employeesTable. Można też kliknąć prawym przyciskiem myszy tabelę employeesTable i wybrać opcję **Właściwości powiązań**.
3. Ponieważ w aplikacji nie ma żadnych obiektów danych, należy taki obiekt dodać. Kliknij opcję **Nowy obiekt danych źródła danych**.
4. W polu **Typ źródła** wybierz opcję **Usługa Web Service**.
5. Ponieważ do aplikacji nie zostało jeszcze dodane źródło danych usługi Web Service, należy je teraz dodać. Obok pola **Źródło danych** kliknij przycisk ..., aby otworzyć okno dialogowe dodawania źródła danych usługi Web Service, które wyszuka w projekcie dostępne klienty usługi Web Service, czyli serwery proxy.
6. Wybierz usługę Web Service EmployeeDirectory i kliknij przycisk Zakończ. Do pliku DirectoryApp.java zostanie dodane nowe źródło danych.

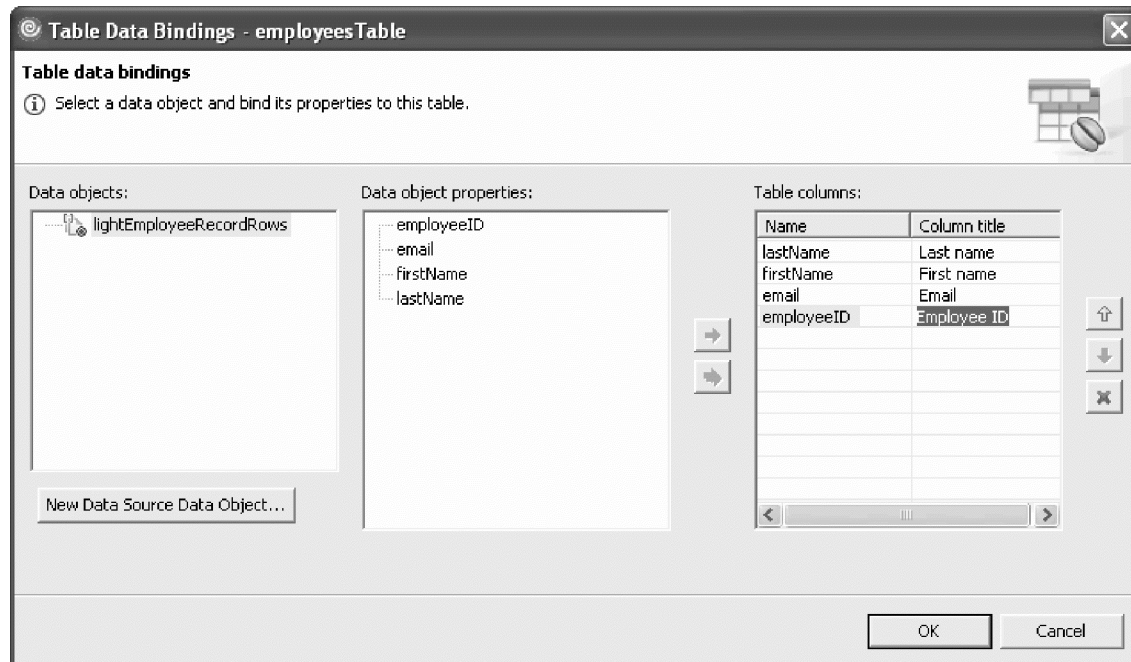


7. W oknie dialogowym nowego obiektu danych źródła danych wybierz metodę getLightEmployeeRecords() w polu Usługa źródła i zaakceptuj domyślną nazwę nowego obiektu danych: lightEmployeeRecordRows. Ta metoda usługi nie wymaga żadnych parametrów. Kliknij przycisk OK. Nowy obiekt danych zostanie utworzony i wyświetlony w wolnym obszarze formularza w widoku projektowania.



Wskazówka: Ponieważ wykonywane jest wiązanie tabeli, w oknie dialogowym Nowy obiekt danych źródła danych wyświetlane są tylko usługi zwracające obiekty danych wiersza. W tym przypadku metoda `getLightEmployeeRecords()` jest jedyną dostępną usługą zwracającą tablicę obiektów.

8. W oknie dialogowym Powiązania danych tabeli wybierz obiekt danych `lightEmployeeRecordRows`.
9. Teraz należy wybrać właściwości obiektu danych `lightEmployeeRecordRows`, który ma być wyświetlany w tabeli `employeesTable`:



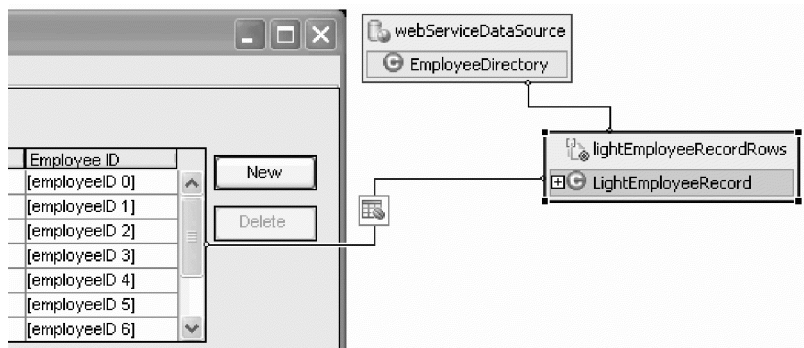
- a. Kliknij przycisk podwójnej strzałki ⇄, aby dodać wszystkie właściwości obiektu do listy **Kolumny tabeli**.
- b. Użyj strzałek w górę i w dół, aby uporządkować kolumny od góry do dołu w następujący sposób: `lastName`, `firstName`, `email`, `employeeID`.

- c. Zmień tytuły kolumn: Nazwisko, Imię, Adres e-mail, Identyfikator pracownika.

Wskazówka: Po zakończeniu wiązania tabeli można zawsze wrócić do właściwości powiązania i zmienić nazwy oraz kolejność kolumn.

- d. Kliknij przycisk **OK**.

Tabela `employeesTable` jest teraz powiązana z obiektem danych `lightEmployeeRecordRows` za pomocą obiektu `JRowTableBinder`. W przypadku kliknięcia obiektu danych `lightEmployeeRecordRows` w wolnym obszarze formularza, edytor wizualny narysuje linię od obiektu danych do tabeli. Na tej linii obiekt `JRowTableBinder` jest reprezentowany przez ikonę konsolidatora tabeli. Kolejna linia wskazuje, że obiekt danych używa obiektu `webServiceDataSource` jako swojego źródła danych.



Punkt kontrolny lekcji

Zwróć uwagę na zmiany w projekcie i aplikacji. W trakcie tej lekcji użytkownik dodał źródło danych usługi Web Service, obiekt danych wiersza i konsolidator wiążący tabelę `employeesTable` z obiektem danych wiersza.

Sprawdź nowy pakiet (`jve.generated`) utworzony w projekcie w celu przechowywania wszystkich klas konsolidatora wygenerowanych przez wizualny edytor Java. Zwróć również uwagę na nowy pakiet (`directory.service`) przechowujący serwer proxy Java usługi Web Service. Opisz lub podsumuj wiedzę zdobytą podczas tej lekcji.



Po uruchomieniu aplikacji Firmowa książka adresowa tabela pracowników zostanie teraz wypełniona przez usługę Web Service istniejącymi rekordami pracowników.

Lekcja 2.3: Powiązanie pól szczegółów z wyborem w tabeli

W poprzednim ćwiczeniu tabela `employeesTable` została powiązana z obiektem danych `lightEmployeeRecordRows` zwracany przez usługę `getLightEmployeeRecords()` w usłudze Web Service. Teraz należy wypełnić pola szczegółów na podstawie pracownika wybranego w tabeli.

Aby uzyskać dodatkowe szczegóły na temat każdego wybranego pracownika, należy użyć kolejnego obiektu danych. Obiekt danych `selectedEmployeeRecord`, który zostanie dodany, jest zwracany przez usługę `getFullEmployeeRecord()`. Ta usługa pobiera identyfikator pracownika wybranego w tabeli jako parametr, a następnie pobiera dodatkowe informacje dotyczące pracownika włącznie z numerem telefonu i położeniem biura.

Obiekt `JRowTableBinder` używany podczas wiązania tabeli z obiektem danych wiersza upraszcza tę czynność. Obiekt `JRowTableBinder` prezentuje element wybrany w tabeli jako oddzielny obiekt danych, którego można użyć jako parametru dla metody `getFullEmployeeRecord(java.lang.Integer)`. Następnie można łatwo powiązać każde z pól tekstowych z odpowiadającą mu właściwością w obiekcie danych `selectedEmployeeRecord`.

Dowiedz się więcej o tej usłudze Web Service: Usługa Web Service obejmuje dwie usługi do pobierania wszystkich szczegółów każdego pracownika. W tabeli są przechowywani wszyscy pracownicy, ale wyświetlany jest tylko pewien podzbiór danych. Po wybraniu jednego pracownika można pobrać pozostałe informacje dotyczące tylko tego pracownika. Gdyby usługa Web Service wysyłała wszystkie dane o każdym pracowniku, gdy tabela zażąda danych, ruch w sieci WWW byłby bardzo duży, co mogłoby zmniejszyć wydajność aplikacji.

Jeśli na przykład rekord pracownika zawiera zdjęcie lub załącznik, nie ma potrzeby pobierania wszystkich zdjęć podczas zwykłego przeglądania pełnej listy pracowników. W związku z tym usługa `getLightEmployeeRecord` służy do zapełniania tabeli, a usługa `getFullEmployeeRecord` pobiera pełny rekord pracownika wybranego w tabeli.

Powiązanie pola nazwiska

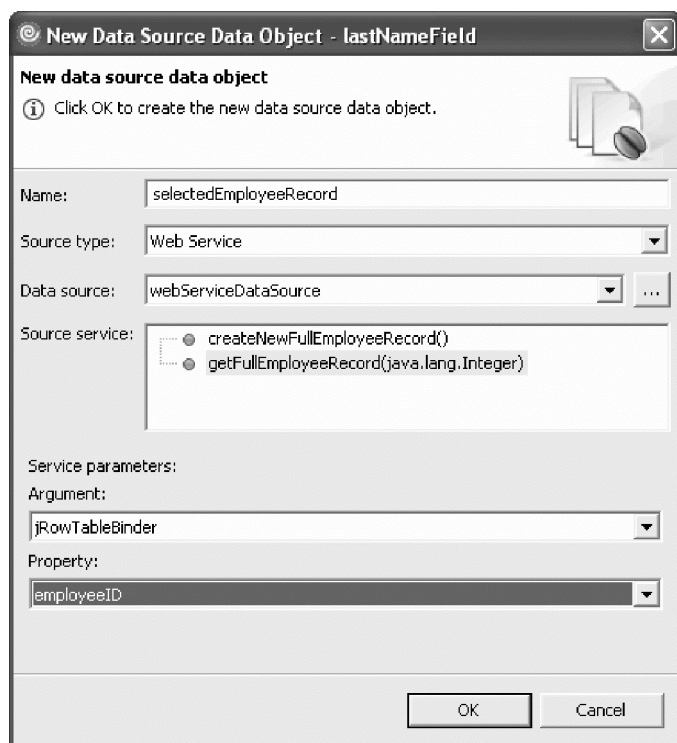
W tym kroku użytkownik powiąże pole **Nazwisko** z właściwością `lastName` obiektu danych `selectedEmployeeRecord`:

1. W widoku komponentów bean Java lub widoku projektowania wybierz pole `TextField` nazwiska (`lastNameField`). W obszarze projektowania na polu tekstowym zostanie wyświetlona zakładka **Wiązanie**.

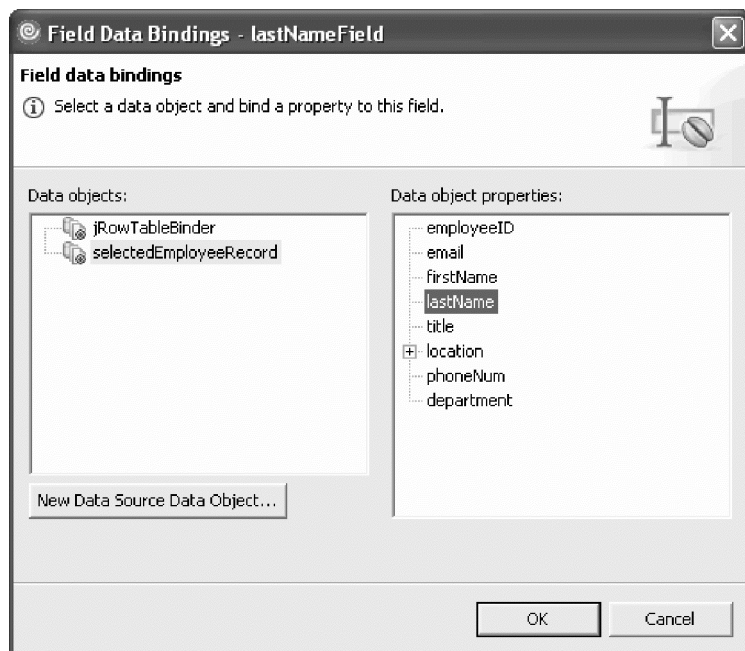


2. Kliknij zakładkę **Wiązanie**, aby otworzyć okno dialogowe Powiązania danych pól.
3. Kliknij opcję **Nowy obiekt danych źródła danych**. Chociaż istniejący obiekt danych `TableRowTableBinder` zwraca poprawne nazwisko, nie obejmuje on pełnego rekordu pracownika. Należy utworzyć nowy obiekt danych reprezentujący pełny rekord pracownika.
4. W polu **Typ źródła** upewnij się, że wybrana jest opcja **Usługa Web Service**, a w polu **Źródło danych** upewnij się, że wybrana jest opcja **webServiceDataSource**.
5. Na liście **Usługa źródła** wybierz metodę `getFullEmployeeRecord(java.lang.Integer)`. Okno dialogowe Nowy obiekt danych źródła danych zawiera usługi zwracające obiekty danych zgodne z polem tekstowym.
6. W polu **Nazwa** wpisz `selectedEmployeeRecord`.
7. W polu **Argument** wybierz wartość `TableRowTableBinder`, a w polu **Właściwość** wybierz wartość `employeeID`. Identyfikator pracownika wybranego wiersza jest teraz ustawiony jako argument metody usługi `getFullEmployeeRecord()`.

Uwaga: Metoda `getFullEmployeeRecord(java.lang.Integer)` wymaga podania liczby całkowitej jako argumentu. Do pobrania pełnego rekordu należy użyć identyfikatora pracownika dla bieżącego wyboru w tabeli pracowników. Podczas wiązania tabeli edytor wizualny automatycznie generuje obiekt `TableRowTableBinder` nasłuchujący bieżącego wyboru w tabeli pracowników. Jako parametru całkowitego należy użyć wartości `employeeID` z wybranego wiersza w obiekcie `TableRowTableBinder`.



8. Kliknij przycisk **OK**.
9. W oknie dialogowym Powiązania danych pól upewnij się, że obiekt `selectedEmployeeRecord` został wybrany na liście **Obiekty danych**. Należy zauważyć, że więcej dostępnych właściwości istnieje dla obiektu danych `selectedEmployeeRecord` niż dla obiektu danych `jRowTableBinder`.
10. Na liście **Właściwości obiektu danych** wybierz właściwość `lastName`.



11. Kliknij przycisk **OK**. Pole nazwiska w aplikacji zostało teraz powiązane z właściwością `lastName` obiektu danych `selectedEmployeeRecord` zwracanego przez usługę `getFullEmployeeRecord()`.
Nowy obiekt danych o nazwie `selectedEmployeeRecord` został utworzony i dodany do aplikacji. Wizualna reprezentacja obiektu danych została dodana do wolnego obszaru formularza w widoku projektowania, co zostało pokazane na poniższym obrazku:



Teraz, po wybraniu pola lastName w obszarze projektowania, linia wskazuje, że jest ono powiązane z obiektem selectedEmployeeRecord. Na środku linii ikona konsolidatora tekstu reprezentuje obiekt SwingTextComponentBinder użyty do tego wiązania. Po wybraniu linii lub ikony reprezentującej konsolidator w obszarze projektowania można sprawdzić właściwości konsolidatora w widoku Właściwości.

Powiązanie pozostałych pól szczegółów

Aby powiązać każde z pozostałych pól szczegółów pracownika, należy postępować w analogiczny sposób, co dla pola nazwiska, ale bez potrzeby dodawania obiektu danych. Ponieważ obiekt danych selectedEmployeeRecord został już dodany, wystarczy tylko powiązać każde pole z odpowiadającą mu właściwością w obiekcie danych selectedEmployeeRecord.

Aby powiązać pola, należy wykonać poniższe czynności dla każdego z pól w sekcji Informacje o pracowniku aplikacji:

1. Wybierz pole w widoku projektowania i kliknij zakładkę **Wiązanie**.
2. W oknie dialogowym Powiązania danych pól wybierz obiekt selectedEmployeeRecord z listy **Obiekty danych**.
3. Na liście **Właściwości obiektu danych** wybierz odpowiednią właściwość dla wiążanego pola. Na poniższym wykresie zostały pokazane właściwości, z którymi należy powiązać poszczególne pola tekstowe:

Pole	Właściwość w obiekcie danych selectedEmployeeRecord
lastNameField	lastName
firstNameField	firstName
idField	employeeID
emailField	email
phoneField	phoneNum
officeField	location.office
buildingField	location.building
siteField	location.site

4. Kliknij przycisk **OK**.

Po zakończeniu wiązania pól tekstowych obszar projektowania powinien wyglądać podobnie do przedstawionego na poniższym obrazku:

Określanie pola identyfikatora pracownika jako pola tylko do odczytu

Pole identyfikatora pracownika jest wyłączone, ponieważ właściwość możliwości edycji tego pola została ustawiona na wartość `false`. Jednak domyślne zachowanie konsolidatora pola tekstowego zmienia stan włączenia pola, jeśli obiekt danych zawiera wartość. Można wyłączyć takie zachowanie konsolidatora, aby pole pozostawało w swoim początkowym stanie tylko do odczytu.

Aby uniemożliwić konsolidatorowi automatyczną zmianę właściwości określającej możliwość edycji:

1. Wybierz pole identyfikatora pracownika. W obszarze projektowania zostanie wyświetlona linia z ikoną  reprezentującą konsolidator pola.
2. Kliknij ikonę konsolidatora  dla pola identyfikatora pracownika.
3. W widoku Właściwości zmień wartość właściwości `autoEditable` na **false**. Naciśnij klawisz **Enter**.

Punkt kontrolny lekcji

Teraz, po uruchomieniu aplikacji i wybraniu pracownika w tabeli, w polach szczegółów zostaną wyświetlone szczegółowe informacje z rekordu danego pracownika.

Lekcja 2.4: Powiązanie przycisku Aktualizacja z konsolidatorem działań

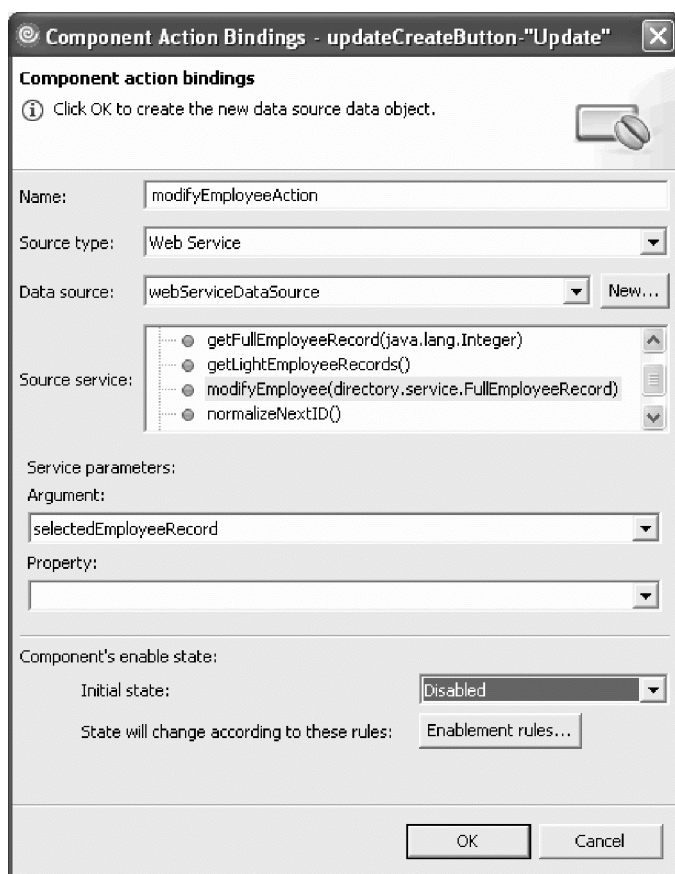
Wizualny edytor Java udostępnia konsolidatory czynności w celu wywoływania usługi źródła danych po kliknięciu przycisku. Jeśli na przykład zostanie kliknięty przycisk Aktualizacja, aplikacja powinna uruchomić metodę `modifyEmployee()` usługi Web Service ze zmianami wprowadzonymi w polach szczegółów. W tej lekcji użytkownik powiąże przycisk Aktualizacja z konsolidatorem czynności.

Aby powiązać przycisk Aktualizacja:

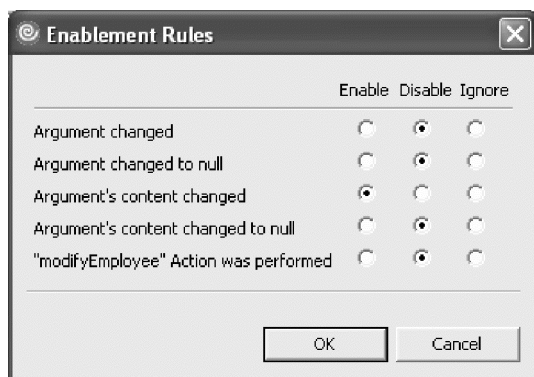
1. W obszarze projektowania wybierz przycisk **Aktualizacja** i kliknij zakładkę **Wiązanie**, aby otworzyć okno dialogowe Powiązania działań komponentów.



2. W polu **Typ źródła** wybierz opcję **Usługa Web Service**.
3. W polu **Źródło danych** wybierz obiekt **webServiceDataSource**.
4. Na liście **Usługa źródła** wybierz metodę **modifyEmployee(directory.service.FullEmployeeRecord)**.
5. Pole **Nazwa** automatycznie zmieni wartość na **modifyEmployeeAction**. Zaakceptuj tę wartość domyślną.
6. W polu **Argument** wybierz pozycję **selectedEmployeeRecord**.
7. Ponieważ metoda `modifyEmployee()` przyjmuje jako argument pełny rekord pracownika, pole **Właściwość** musi pozostać puste.
8. Ustaw wartość **Stan początkowy** przycisku na **Wyłączony**.



9. Aby zdefiniować zmiany stanu przez przycisk, kliknij przycisk **Reguły aktywowania**. Określ, że przycisk jest włączony tylko w przypadku zmiany zawartości argumentu i wyłączony we wszystkich pozostałych przypadkach. Kliknij przycisk **OK**.



Oznacza to, że przycisk **Aktualizacja** jest wyłączony do chwili wprowadzenia zmiany w obiekcie `selectedEmployeeRecord`. Inaczej mówiąc, konsolidator włączy ten przycisk po wpisaniu nowej wartości w jednym z pól szczegółów powiązanych z obiektem `selectedEmployeeRecord`. Po wybraniu nowego rekordu lub kliknięciu przycisku **Aktualizacja** przycisk zostanie ponownie wyłączony.

10. Kliknij przycisk **OK**.

Nowy konsolidator `SwingDataServiceAction` zostanie dodany dla przycisku **Aktualizacja**. Po wybraniu przycisku w obszarze projektowania edytor wizualny narysuje linię oznaczającą, że przycisk jest powiązany ze źródłem danych usługi `Web Service`. Różowa, kropkowana strzałka jest skierowana od obiektu `selectedEmployeeRecord` do linii. Ta strzałka wskazuje, że obiekt `selectedEmployeeRecord` jest argumentem wywołania usługi.

Punkt kontrolny lekcji

Teraz po uruchomieniu aplikacji można będzie zaktualizować rekord pracownika.

Wybierz pracownika w tabeli i zmień jego nazwisko. Jak tylko nazwisko zostanie zmienione, włączony zostanie przycisk **Aktualizacja**. Po kliknięciu przycisku **Aktualizacja** zostanie wywołana usługa `modifyEmployee` i dane pracownika zostaną zaktualizowane. Nowe nazwisko zostanie uwzględnione w tabeli pracowników.

Lekcja 2.5: Włączenie przycisku Usuń i okna dialogowego potwierdzenia

W tym ćwiczeniu użytkownik zaprogramuje aplikację Firmowa książka adresowa tak, aby można było usunąć rekord pracownika.

Poniższa lista opisuje zachowania, których ma używać aplikacja:

- Po wybraniu pracownika w tabeli zostanie włączony przycisk **Usuń**.
- Po kliknięciu przycisku **Usuń** zostanie otwarte okno dialogowe Potwierdź usunięcie z pytaniem o potwierdzenie usunięcia.
- Jeśli użytkownik kliknie przycisk **Tak** w oknie dialogowym Potwierdź usunięcie, rekord pracownika zostanie usunięty, okno Potwierdź usunięcie zostanie zamknięte, a lista pracowników zostanie odświeżona.
- W przypadku kliknięcia przycisku **Nie** usunięcie zostanie anulowane, a okno dialogowe Potwierdź usunięcie zostanie zamknięte.

Programowanie włączania i wyłączania przycisku Usuń w zależności od tego, czy w tabeli został wybrany wiersz

Aby zaprogramować włączanie i wyłączanie przycisku **Usuń**, należy do tabeli dodać procedurę nasłuchiwaną włączającą przycisk po wybraniu wiersza.

1. Wybierz tabelę `employeesTable` w widoku komponentów bean Java. W widoku źródła zostanie podświetlony następujący wiersz:

```
employeesTable = new JTable();
```
2. Bezpośrednio za tym wierszem dodaj nowy obiekt `ListSelectionListener` i zdarzenie `valueChanged` do obiektu `employeesTable`:

```
employeesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        getDeleteButton().setEnabled(getEmployeesTable().getSelectedRowCount() != 0);  
    }  
});
```
3. Po dodaniu tych wierszy kodu edytor źródła oznaczy je jako błędy do chwili zaimportowania obiektu `ListSelectListener` i `ListSelectionEvent`. Aby dodać wymagane instrukcje importu, w menu głównym kliknij opcję **Źródło → Uporządkuj instrukcje importu**. Do sekcji instrukcji importu klasy zostaną dodane następujące wiersze:

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

Teraz po wybraniu wiersza w tabeli przycisk **Usuń** zostanie włączony.

Programowanie otwierania okna dialogowego Potwierdź usunięcie po kliknięciu przycisku Usuń

Dodaj zdarzenie `actionPerformed` do przycisku **Usuń** i zaprogramuj je do otwierania okna dialogowego Potwierdź usunięcie.

1. Kliknij prawym przyciskiem myszy przycisk **Usuń** i wybierz opcję **Zdarzenia → actionPerformed**. Do metody `getDeleteButton()` zostanie dodany następujący szkielec zdarzenia:

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        System.out.println("actionPerformed()");  
        // TODO Automatycznie wygenerowany kod pośredniczący zdarzenia actionPerformed()  
    }  
});
```

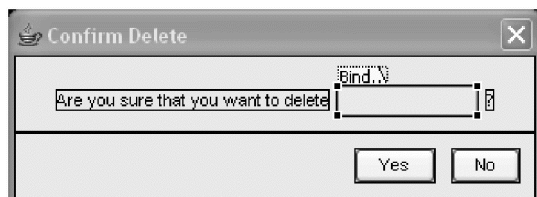

2. Zastąp ten wygenerowany szkielet poniższym kodem, który włącza widoczność okna dialogowego Potwierdź usunięcie po kliknięciu przycisku:

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

Wiązanie pola tekstowego w oknie dialogowym Potwierdź usunięcie

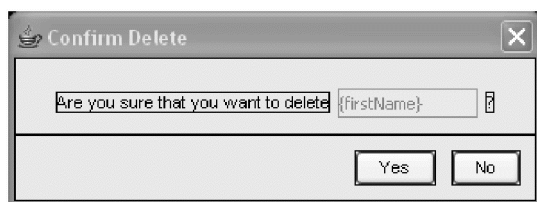
Powiązanie pole tekstowe w oknie dialogowym Potwierdź usunięcie, aby wyświetlić imię pracownika do usunięcia.

1. W widoku komponentów bean Java lub obszarze projektowania wybierz pole tekstowe employeeToDeleteField i kliknij zakładkę **Wiązanie**.



2. W oknie dialogowym Powiązania danych pól wybierz obiekt danych selectedEmployeeRecord i pole firstName, a następnie kliknij przycisk **OK**.

Pole tekstowe jest teraz powiązane z kolumną firstName wiersza wybranego w tabeli employeesTable.



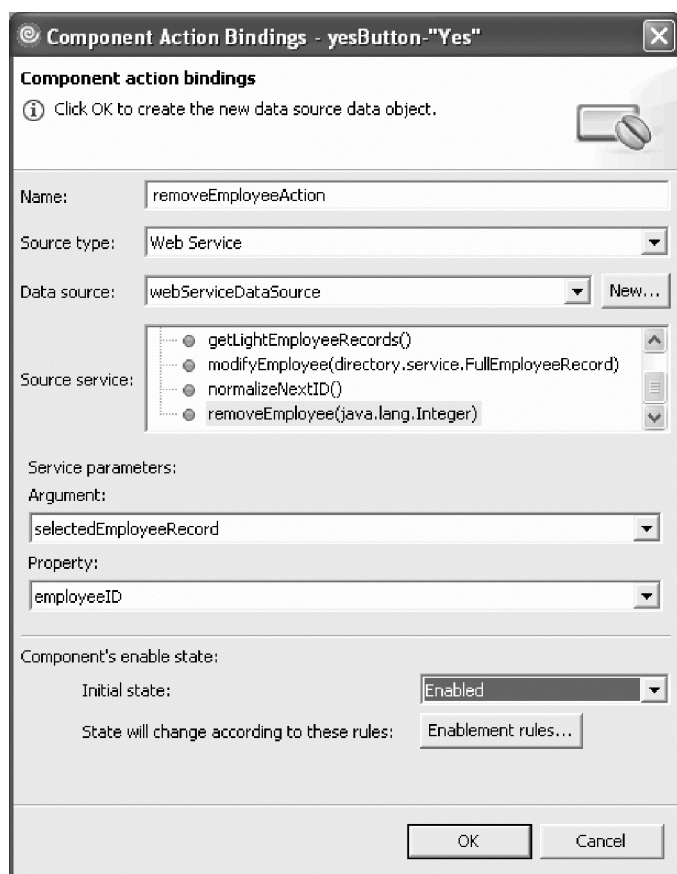
3. Aby zapewnić, że pole to jest tylko do odczytu, ustaw właściwość **autoEditable** konsolidatora pola na wartość **false**.

Wiązanie przycisku Tak do wykonywania operacji usunięcia

Powiązanie przycisk **Tak** tak, aby wywoływał metodę removeEmployee(java.lang.Integer) usługi Web Service.

1. Wybierz przycisk **Tak** i kliknij zakładkę **Wiązanie**, aby otworzyć okno dialogowe Powiązania działań komponentów.
2. W polu **Typ źródła** wybierz opcję **Usługa Web Service**.
3. W polu **Źródło danych** wybierz obiekt **webServiceDataSource**.
4. Na liście **Usługa źródła** wybierz metodę **removeEmployee(java.lang.Integer)**.
5. Pole **Nazwa** automatycznie zmieni wartość na **removeEmployeeAction**. Zaakceptuj tę wartość domyślną.
6. W polu **Argument** wybierz pozycję **selectedEmployeeRecord**.
7. W polu **Właściwość** wybierz pozycję **employeeID**. Ponieważ metoda removeEmployee() przyjmuje jako argument liczbę całkowitą, należy użyć identyfikatora pracownika obiektu selectedEmployeeRecord.
8. Ustaw wartość **Stan początkowy** przycisku na **Włączony**.
9. W polu **Reguły aktywowania** wybierz opcję **Ignoruj** dla każdego z warunków.

Taki stan komponentu oznacza, że przycisk Tak będzie zawsze włączony, ponieważ nie ma potrzeby zmiany jego stanu.



10. Kliknij przycisk OK.

Dodawanie zdarzenia w celu ukrycia okna dialogowego Potwierdź usunięcie po usunięciu pracownika

W tym kroku zostanie dodane zdarzenie do *konsolidatora* przycisku **Tak** (nie do samego przycisku **Tak**). Okno dialogowe Potwierdź usunięcie ma zostać zamknięte po usunięciu pracownika, czyli po pomyślnym wywołaniu usługi źródła danych przez konsolidator.

Dodaj do metody `getRemoveEmployeeAction()` następujący kod:

```
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) {}
});
```

Ten kod zdarzenia ukrywa okno dialogowe Potwierdź usunięcie po wykonaniu operacji przez konsolidator.

Punkt kontrolny lekcji

Teraz po uruchomieniu aplikacji Firmowa książka adresowa można wybrać pracownika w tabeli, kliknąć przycisk **Usuń**, a następnie kliknąć przycisk **Tak** w celu potwierdzenia usunięcia. Rekord pracownika zostanie usunięty z katalogu, co zostanie odzwierciedlone na liście pracowników.

Lekcja 2.6: Skonfigurowanie działań i powiązań w celu dodania nowego pracownika

W tej lekcji użytkownik umożliwi dodanie nowego rekordu pracownika w aplikacji Firmowa książka adresowa.

Ponieważ zachowanie aplikacji jest bardziej skomplikowane i dynamiczne podczas dodawania nowego pracownika, to ćwiczenie jest również bardziej złożone i wymaga ręcznego wprowadzenia pewnych zmian w kodzie źródłowym. Ponadto to ćwiczenie prezentuje pewne zaawansowane możliwości obiektów danych oraz stanowi kreatywny przykład jednego ze sposobów zastosowania konsolidatorów i obiektów danych w celu realizacji konkretnych wymagań.

Poniższa lista opisuje wymagane zachowanie aplikacji:

- Po kliknięciu przycisku **Nowy** zostaną wykonane następujące czynności:
 - Wybór w tabeli pracowników zostanie anulowany i tabela zostanie wyłączona.
 - Anulowanie wyboru w tabeli spowoduje wyłączenie przycisku **Usuń**.
 - Wyłączone zostanie pole **Filtr**.
 - Z pól szczegółów zostaną usunięte wszystkie wartości z wyjątkiem identyfikatora nowego pracownika.
 - Tekst na przycisku **Aktualizacja** zostanie zmieniony na **Dodaj**.
- Po kliknięciu przycisku **Dodaj** zostaną wykonane następujące czynności:
 - Wartości wprowadzone w polach szczegółów zostaną dodane do katalogu jako rekord nowego pracownika.
 - Tabela zostanie włączona, a wartości odświeżone.
 - Zostanie włączone pole **Filtr**.
 - Tekst na przycisku **Dodaj** zostanie ponownie zmieniony na **Aktualizacja**.

Dodawanie nowego obiektu danych źródła danych wywołującego metodę `createNewFullEmployeeRecord()`

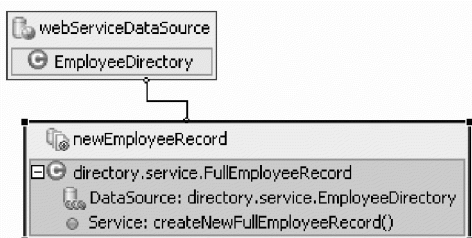
Przykładowa usługa Web Service udostępnia usługę `createNewFullEmployeeRecord` zapewniającą nowy, pusty rekord uzupełniany kolejnym dostępnym numerem identyfikatora pracownika. Ten pusty rekord można następnie wypełnić informacjami o pracowniku i przesłać z powrotem do usługi Web Service.

1. Na palecie wizualnego edytora Java rozwiń szufladę Obiekty danych i wybierz pozycję **Obiekt danych źródła danych**.
2. Przenieś wskaźnik myszy nad pusty obszar widoku projektowania lub wolny obszar formularza i kliknij lewym przyciskiem myszy, aby upuścić obiekt danych źródła danych. Nowy obiekt danych źródła danych zostanie dodany i wyświetlony w wolnym obszarze formularza:



3. Kliknij prawym przyciskiem myszy obiekt danych źródła danych i wybierz opcję **Zmień nazwę pola**. Zmień nazwę obiektu danych na `newEmployeeRecord`.
4. Kliknij prawym przyciskiem myszy obiekt danych `newEmployeeRecord` i wybierz opcję **Właściwości powiązań**. Zostanie otwarte okno dialogowe **Wiązanie danych**.
5. W polu **Źródło danych** wybierz obiekt `webServiceDataSource`.
6. W polu **Usługa** wybierz metodę `createNewFullEmployeeRecord()`.
7. Kliknij przycisk **OK**.

W wolnym obszarze formularza widać, że obiekt danych źródła danych `newEmployeeRecord` jest powiązany z usługą Web Service.



Dodawanie podstawowego obiektu danych w celu ułatwienia przełączania obiektów danych

Ponieważ pola szczegółów i przycisk Aktualizacja wymagają przełączania trybów (w celu dokonania aktualizacji i utworzenia nowego pracownika), wymagają one powiązania z dwoma różnymi obiektami danych w różnym czasie. W celu ułatwienia tej czynności zostanie dodany podstawowy obiekt danych o nazwie `switchingDataObject`. Ten podstawowy obiekt danych będzie służył do przełączania powiązania pól tekstowych między obiektami `selectedEmployeeRecord` i `newEmployeeRecord`.

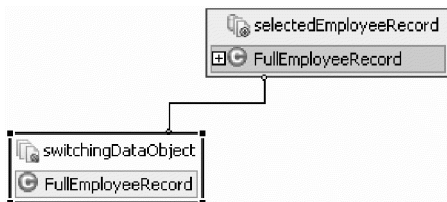
Nowy podstawowy obiekt danych wskazuje po prostu inny obiekt danych (`selectedEmployeeRecord`) zdefiniowany we wcześniejszym ćwiczeniu. Ten nowy obiekt danych będzie przydatny po utworzeniu metody informującej podstawowy obiekt danych o konieczności użycia wcześniej utworzonego obiektu `newEmployeeRecord`. Inaczej mówiąc, podstawowy obiekt danych będzie działał jako pośredni obiekt danych przełączający obiekty danych `selectedEmployeeRecord` i `newEmployeeRecord`, tym samym umożliwiając komponentom wizualnym aplikacji pracę z dwoma różnymi obiektami danych.

1. Na palecie edytora wizualnego wybierz element **Podstawowy obiekt danych** i upuść go w wolnym obszarze formularza. Zostanie dodany obiekt `basicDataObject`.



2. Zmień nazwę obiektu danych na `switchingDataObject`
3. W widoku Właściwości obiektu `switchingDataObject` ustaw właściwość **sourceObject** na wartość **selectedEmployeeRecord**. Wartość `selectedEmployeeRecord` można wybrać z menu rozwijanego w kolumnie Wartość właściwości.

Teraz obiekt `switchingDataObject` odwołuje się do obiektu `selectedEmployeeRecord` i zawiera te same wartości:

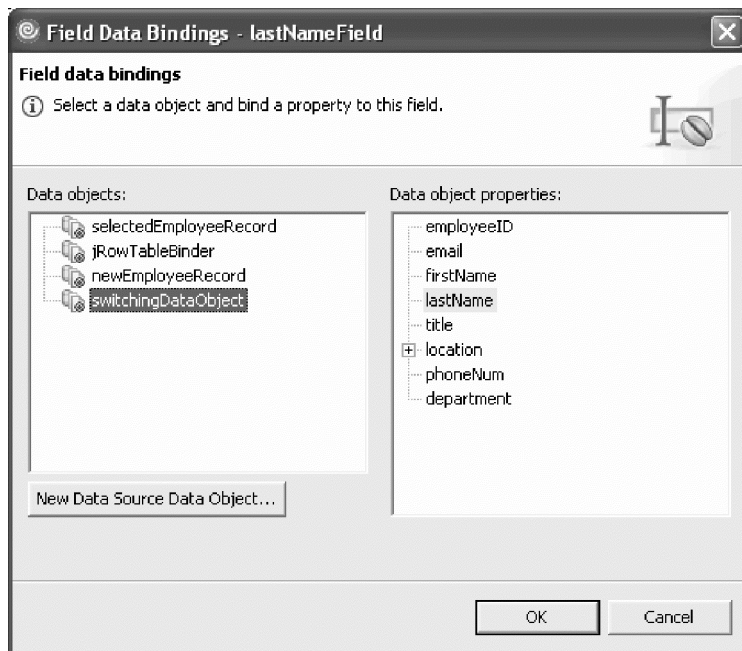


Ponowne wiązanie każdego pola pracownika z obiektem `switchingDataObject`

Chociaż każde z pól szczegółów pracownika jest już powiązane z obiektem `selectedEmployeeRecord`, należy teraz powiązać je z obiektem `switchingDataObject`. Po powiązaniu pól można dynamicznie przełączać się między obiektami danych tych pól w zależności od tego, czy następuje modyfikacja istniejącego rekordu pracownika czy też dodawany jest nowy rekord pracownika.

Dla każdego pola w sekcji Informacje o pracowniku należy wykonać poniższe czynności:

1. Wybierz pole i kliknij zakładkę **Wiązanie**.
2. W oknie dialogowym Powiązania danych pól wybierz obiekt `switchingDataObject`. Wcześniej pola te zostały powiązane z obiektem `selectedEmployeeRecord`.



- Upewnij się, że pole jest nadal powiązane z poprawną właściwością obiektu danych i kliknij przycisk **OK**. Po wybraniu pola w widoku projektowania widać będzie, że linie konsolidatora wskazują teraz obiekt `switchingDataObject`.



Definiowanie flagi oraz metody aktualizacji i przełączania trybów

Poniższa metoda `updateMode()` sprawdza, czy flaga trybu jest ustawiona na tworzenie nowego rekordu, a następnie odpowiednio zmienia zachowanie aplikacji. Domyślnie flaga boolowska `isNewMode` ma wartość `false`, a metoda `updateMode()` włącza tabelę pracowników i pole filtru oraz ustawia tekst przycisku Aktualizuj na "Aktualizuj". Jeśli flaga `isNewMode` ma wartość `true`, tabela pracowników jest wyłączana, są anulowane wszystkie wybory, jest wyłączane pole filtru, a tekst na przycisku Aktualizuj jest zmieniany na "Dodaj".

Poniższy kod należy dodać do klasy `DirectoryApp.java` tuż przed ostatnim zamykającym nawiasem klamrowym:

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

Dodawanie zdarzenia `actionPerformed` do przycisku Nowy

W tym kroku zostanie dodany kod zdarzenia wywoływanego po kliknięciu przycisku **Nowy**. To zdarzenie informuje obiekt `switchingDataObject` o konieczności użycia obiektu danych `newEmployeeRecord`, ustawia flagę trybu na "new" i uruchamia metodę `updateMode()` dodaną w poprzednim kroku.

- W widoku projektowania kliknij prawym przyciskiem myszy przycisk **Nowy** i wybierz opcję **Zdarzenia** → **actionPerformed**. W metodzie `getNewButton()` zostanie wygenerowany następujący kod:


```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Automatycznie wygenerowany kod pośredniczący zdarzenia actionPerformed()
    }
});
```

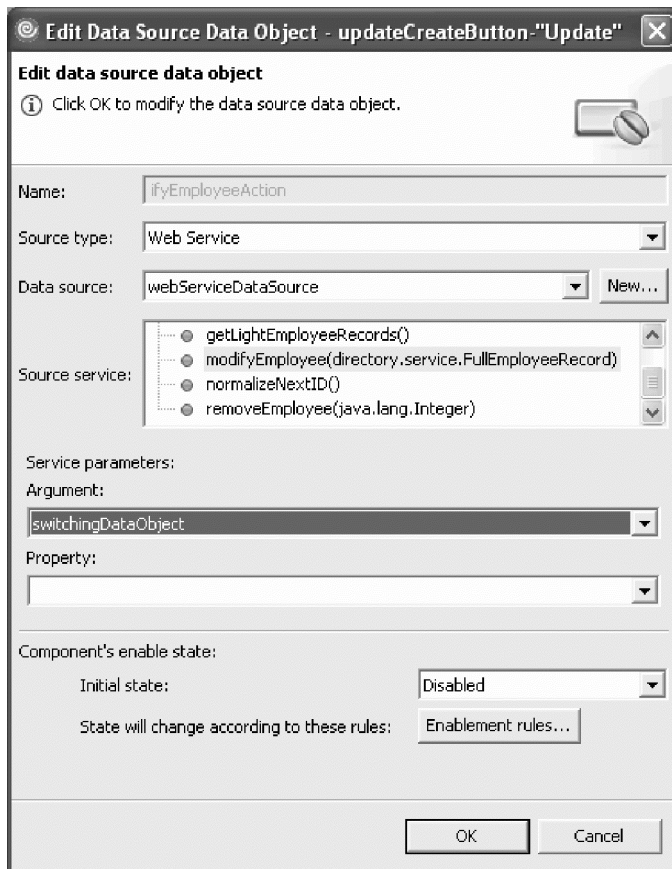
2. Zastąp ten wygenerowany szkielet poniższym kodem:

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getSwitchingDataObject().setSourceObject(getNewEmployeeRecord());
        getNewEmployeeRecord().refresh();
        isNewMode = true; // przełącza aplikację w tryb nowego rekordu
        updateMode(); // zmienia interfejs użytkownika odpowiednio do nowego trybu
        getLastNameField().grabFocus();
    }
});
```

Ponowne wiązanie przycisku Aktualizacja

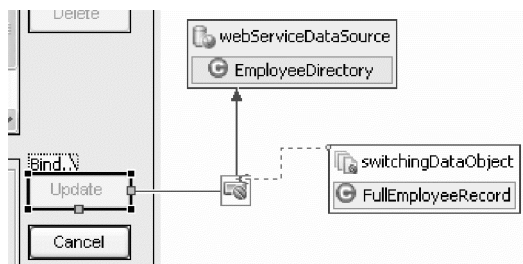
W poprzedniej lekcji przycisk **Aktualizacja** został zaprogramowany do korzystania z metody `modifyEmployee` usługi Web Service. Ta czynność jest zaimplementowana jako `SwingDataServiceAction`. Jedną z właściwości czynności `SwingDataServiceAction` jest obiekt źródłowy działający jako argument usługi. Obiekt źródłowy dla czynności modyfikacji jest obecnie ustawiony na obiekt `selectedEmployeeRecord`. Aby zaprogramować ten przycisk do sterowania zarówno aktualizacją, jak i dodawaniem, należy ponownie skonfigurować czynność tego przycisku w celu używania obiektu `switchingDataObject` jako argumentu dla usługi `modifyEmployee`.

1. W widoku projektowania wybierz przycisk **Aktualizacja**. Zwróć uwagę na różową, kropkowaną strzałkę wskazującą, że obiekt `selectedEmployeeRecord` jest argumentem wywołania usługi.
2. Kliknij zakładkę **Wiązanie** dla przycisku **Aktualizacja**.
3. W polu **Argument** wybierz pozycję `switchingDataObject`.



4. Kliknij przycisk **OK**.

Należy teraz zauważyć, że czynność przycisku została teraz skonfigurowana do korzystania z obiektu `switchingDataObject` jako argumentu dla metody `modifyEmployee`:



Dodawanie zdarzenia do konsolidatora przycisku Aktualizacja na potrzeby resetowania trybu

Po kliknięciu przycisku **Aktualizacja** i zakończeniu czynności w usłudze Web Service aplikacja powinna powrócić do swojego domyślnego trybu i zachowania. W tym celu należy dodać procedurę nasłuchiwanie zdarzeń w konsolidatorze działań związanych z przyciskiem, która będzie aktualizować tryb i odświeżać tabelę po zakończeniu aktualizacji lub dodawania.

Dla przycisku Aktualizacja dodaj do metody `getModifyEmployeeAction()` następujący kod:

```
modifyEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        if (isNewMode) {
            // wróć do używania obiektu selectedEmployeeRecord
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            // wycofaj tryb nowego rekordu
            isNewMode = false;
            updateMode();
        }
        // odśwież obiekt danych tabeli
        getLightEmployeeRecordRows().refresh();
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

Punkt kontrolny lekcji

Po uruchomieniu aplikacji Firmowa książka adresowa można teraz kliknąć przycisk **Nowy** i dodać rekord nowego pracownika.

Lekcja 2.7: Programowanie działania przycisku Anuluj

Podczas korzystania z aplikacji wymagana jest możliwość łatwego wycofania wszelkich rozpoczętych zmian w rekordzie pracownika, jeśli zmiany te nie mają zostać wprowadzone. Inaczej mówiąc musi istnieć możliwość anulowania i wyczyszczenia pól, aby można było rozpocząć od początku. Aby dodać tę funkcję, należy ustawić pewne zdarzenia `actionPerformed` dla przycisku **Anuluj**.

Poniższa lista opisuje wymagane zachowanie przycisku **Anuluj**:

- W przypadku kliknięcia przycisku **Anuluj** w trybie nowego rekordu aplikacja zakończy ten tryb.
- Jeśli przycisk **Anuluj** zostanie kliknięty podczas modyfikowania rekordu pracownika, wszelkie zmienione wartości zostaną wycofane i przywrócone zostaną pierwotne wartości.

Aby dodać zdarzenie `actionPerformed` do przycisku **Anuluj** w celu wykonania wymaganej operacji:

1. W widoku projektowania kliknij prawym przyciskiem myszy przycisk **Anuluj** i wybierz opcję **Zdarzenia** → **actionPerformed**. W metodzie `getCancelButton()` zostanie wygenerowany następujący kod:

```
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO Automatycznie wygenerowany kod pośredniczący zdarzenia actionPerformed
    }
});
```

2. Zastąp wygenerowany szkielet zdarzenia poniższym kodem:

```
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();
        }
    }
});
```

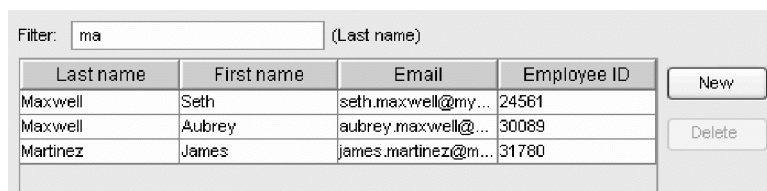
Punkt kontrolny lekcji

W tej lekcji użytkownik dowiedział się o sposobie programowania przycisku **Anuluj** za pomocą zdarzeń `actionPerformed`.

Lekcja 2.8: Skonfigurowanie filtru dla tabeli pracowników

Do filtrowania zawartości tabeli pracowników można użyć konsolidatora filtru tekstowego. Filtr pobiera dane wejściowe z pola tekstowego i filtruje tabelę na podstawie konkretnej właściwości lub kolumny tabeli.

W tej aplikacji znaki wprowadzone w polu **Filtr** będą używane do wykonywania filtrowania według nazwiska pracownika. Jeśli dokładne wartości wprowadzone w polu **Filtr** występują w polu nazwiska rekordu pracownika, rekord pracownika zostanie wyświetlony w tabeli.

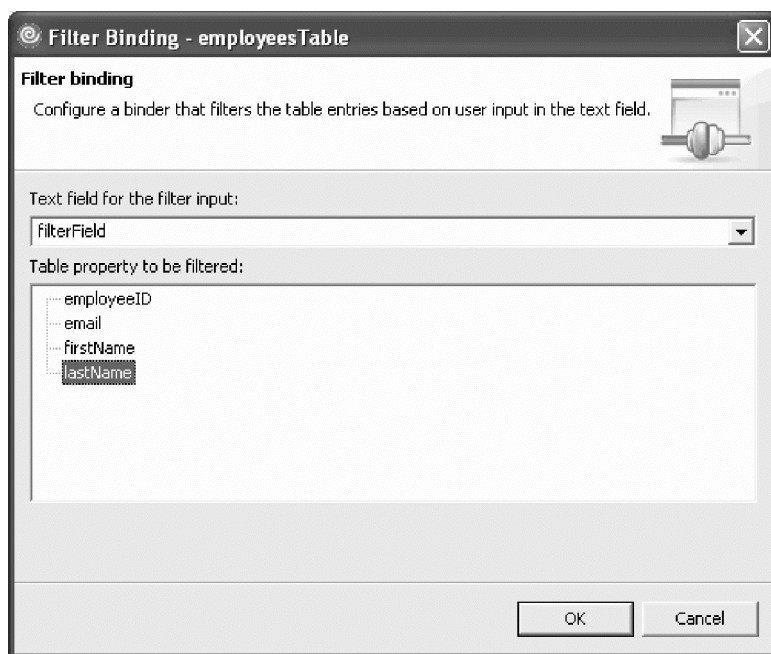


The screenshot shows a web application interface. At the top, there is a text input field labeled "Filter:" containing the text "ma", followed by the text "(Last name)". Below this is a table with four columns: "Last name", "First name", "Email", and "Employee ID". The table contains three rows of data. To the right of the table are two buttons: "New" and "Delete".

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

Aby utworzyć filtr dla tabeli:

1. Wybierz ikonę konsolidatora dla tabeli `employeesTable` i wybierz opcję **Właściwości powiązania filtru**. Zostanie otwarte okno dialogowe **Powiązanie filtru**.
2. Na liście **Pole tekstowe na dane wejściowe filtru** wybierz pozycję `filterField`.
3. Na liście **Właściwości tabeli do filtrowania** wybierz pozycję `lastName`.



4. Kliknij przycisk **OK**.

Zostanie wygenerowany nowy filtr `SwingPropertyFilter`. Właściwość filtru konsolidatora tabeli jest ustawiona na korzystanie z nowego filtru. Nowy filtr jest skonfigurowany do używania pola **Filtr** jako wejścia i do filtrowania na podstawie właściwości `lastName` tabeli.

Punkt kontrolny lekcji

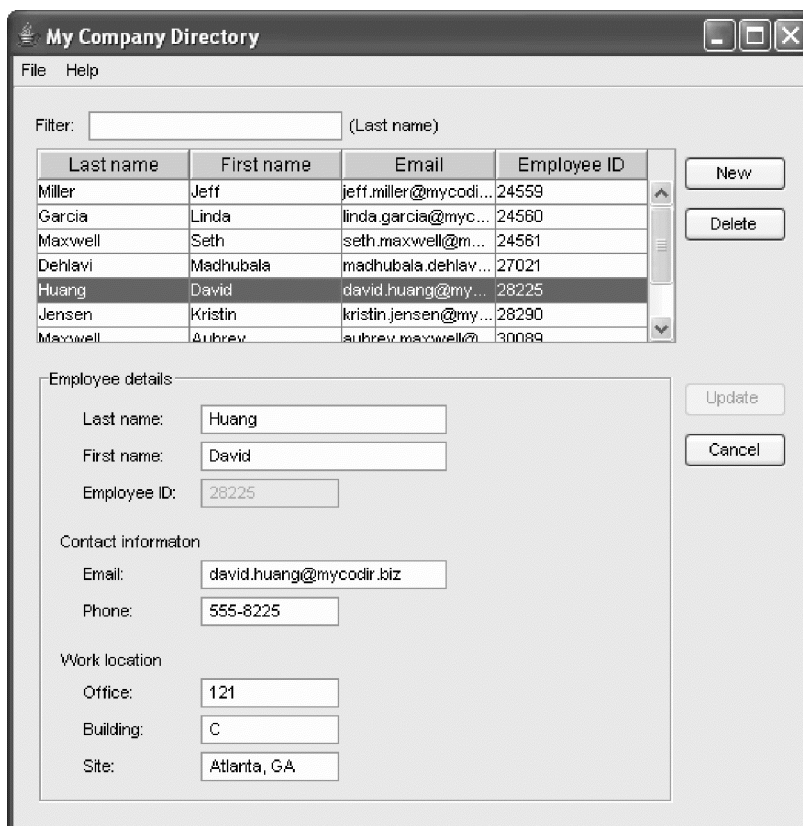
W tej lekcji użytkownik dowiedział się, jak skonfigurować filtr dla tabeli.

Po uruchomieniu aplikacji Firmowa książka adresowa można teraz wpisać znaki w polu **Filtr** i tabela zostanie odfiltrowana w celu wyświetlenia wierszy, w których nazwisko zawiera wprowadzone znaki.

Gratulacje! Aplikacja Firmowa książka adresowa jest gotowa.

Podsumowanie: Budowanie pełnego klienta Java korzystającego z usługi Web Service

Gratulacje! Wiesz już jak korzystać z wizualnego edytora Java, aby zbudować aplikację Firmowa książka adresowa będącą pełnym klientem Java łączącym się z przykładową usługą Web Service w celu obsługi katalogu pracowników.



Patrz obraz ukończonego produktu.:

Zgromadzone doświadczenie

Użytkownika korzystał z edytora wizualnego w celu dokończenia graficznego interfejsu użytkownika, używając obiektu GridBagLayout do określenia układu tabeli pracowników. Następnie tabela, pola i przyciski zostały powiązane z odpowiednimi obiektami danych i źródłem danych w celu umożliwienia współpracy aplikacji z wygenerowanym serwerem proxy Java usługi Web Service. Użytkownik zmodyfikował również kod w celu zapewnienia prawidłowego zachowania aplikacji zapewniającego prostotę i intuicyjność obsługi. Ponadto użytkownik dowiedział się, w jaki sposób zainstalować aplikację korporacyjną na serwerze WebSphere Application Server 6.0 oraz wdrożyć usługę Web Service.

Co ważniejsze, przekazano pełną wiedzę na temat uniwersalnych klas konsolidatora udostępnianych przez wizualny edytor Java na potrzeby pracy z danymi. Użytkownik może teraz rozpocząć samodzielne eksperymenty oraz sprawdzać nowe, interesujące zastosowania konsolidatorów.

Użytkownik powinien teraz umieć wykonać następujące zadania:

- Używanie wizualnego edytora Java do rozmieszczania komponentów w układzie GridBagLayout
- Uruchamianie klasy wizualnej jako komponentu bean Java
- Wiązanie komponentów wizualnych aplikacji Java z metodami i obiektami danych zwracanymi przez usługę Web Service
- Dodawanie zdarzeń do komponentów wizualnych

Dodatkowe zasoby

Zaimportuj ostateczną wersję aplikacji My Directory

W projekcie tym dostępna jest ukończona wersja aplikacji, pakiet `jve.generated` z klasami konsolidatora i klient Java usługi Web Service skonfigurowany dla serwera WebSphere Application Server w wersji 6.1. Jeśli ten ukończony projekt zostanie zaimportowany bez wykonania kroków opisanych w tym kursie, konieczne może być skonfigurowanie zmiennej określającej ścieżkę budowania Java. Ścieżka ta musi wskazywać używany plik JAR cienkiego klienta usług Web Services WebSphere 6.1.

Wskazówka: Jeśli podczas importowania nie zostanie określona inna nazwa projektu, treść projektu MyDirectory użytkownika zostanie zastąpiona.