

DBS – 4. ročník

Programování DB aplikací v jazyce JAVA

Jdeveloper – vzorový příklad

Prostředí JDeveloper a založení projektu

Topologie stránky a navigace

Master / Detail

Interaktivní graf

Hierarchie

Drag & Drop



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

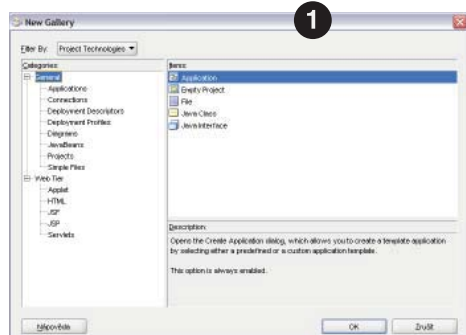
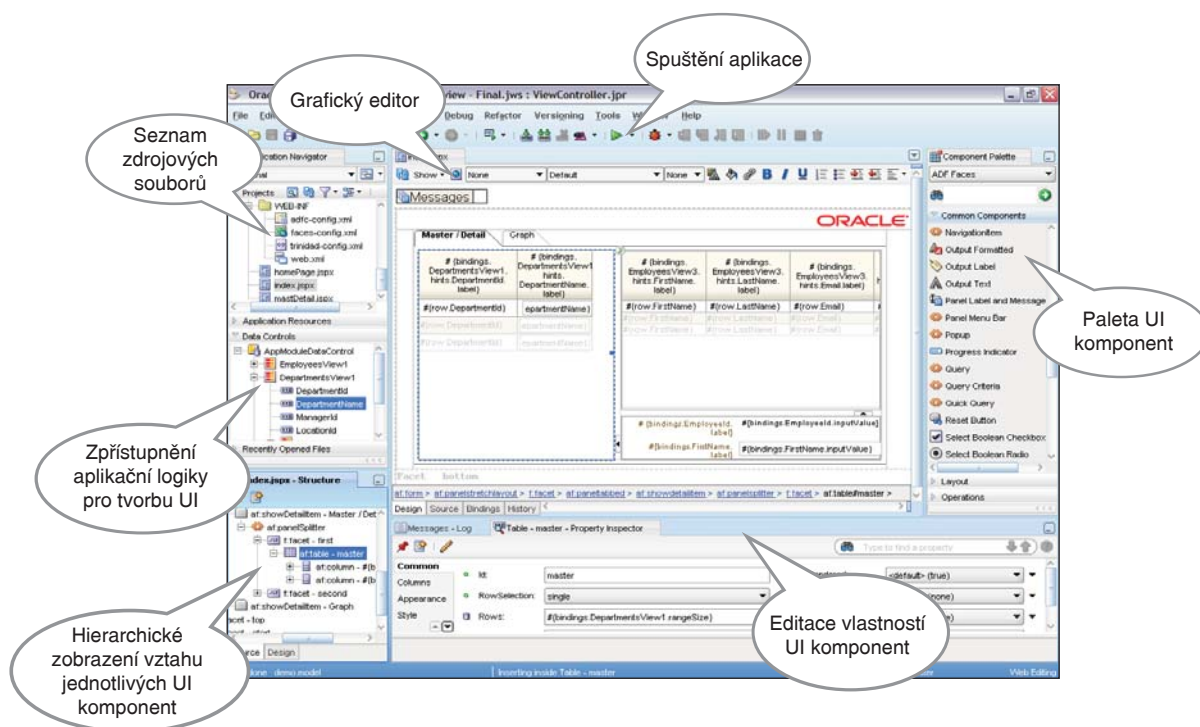


OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Prostředí JDeveloper a založení projektu

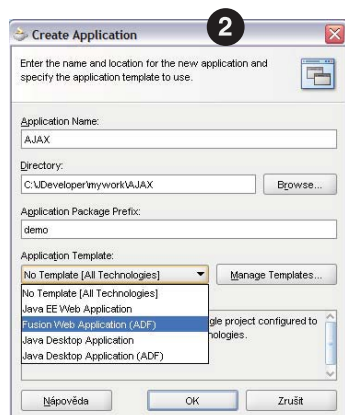
Pro tvorbu ukázky budeme používat vývojové prostředí Oracle JDeveloper 11g, které je zdarma k dispozici na URL adrese www.oracle.com. Instalace nástroje JDeveloper je velmi snadná, stačí extrahovat obsah ZIP archivu a následně spustit soubor `jdeveloper.exe` (platforma Windows). Po úspěšném spuštění nástroje JDeveloper se objeví níže zobrazené pracovní prostředí.



Prvním krokem v naší ukázce bude vytvoření projektu, v rámci kterých budeme budovat jednotlivé ukázky. V menu **File** zvolíme položku **New...** a v zobrazeném dialogu zvolíme položku **Application**. 1 Po této volbě se zobrazí dialog, 2 ve kterém zvolíme název aplikace (např. AJAX) a v položce **Application Template** vybereme variantu **Fusion Web Application(ADF)**.

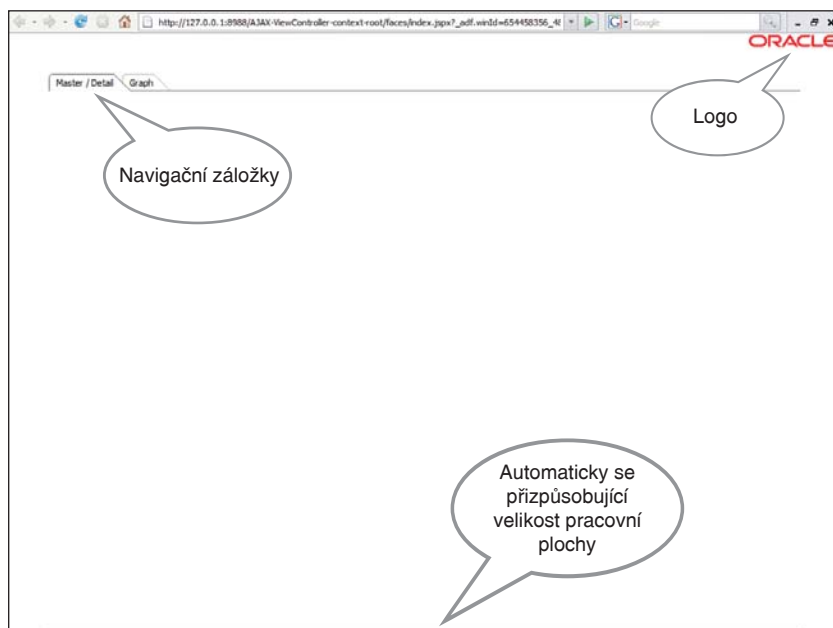
V rámci jednotlivých kapitol nejdříve postupujeme dle uvedených instrukcí a následně spustíme aplikaci a ověříme, zda aplikace odpovídá zadání popsaném v úvodu kapitoly.

V rámci ukázek využijeme dvou předpřipravených obrázků, které najdeme na URL adrese – <http://www.oracle.com/cz/adf>. Obrázky nakopírujeme do adresáře `C:\JDeveloper\mywork\AJAX\ViewController\public_html\images`, s tím že podadresář **images** je nutné vytvořit.



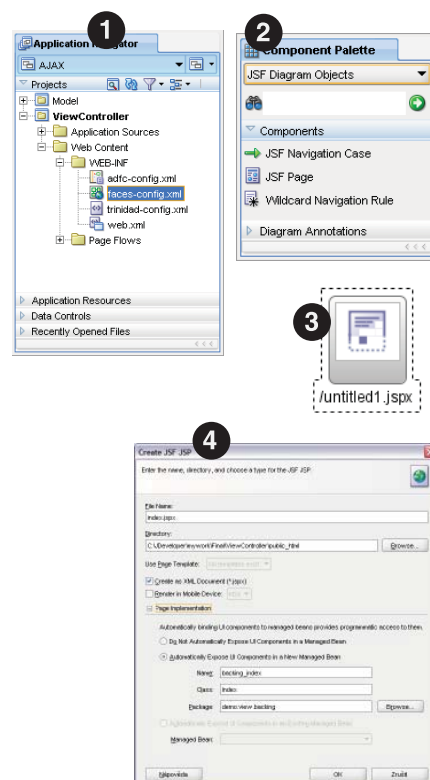
Topologie stránky a navigace

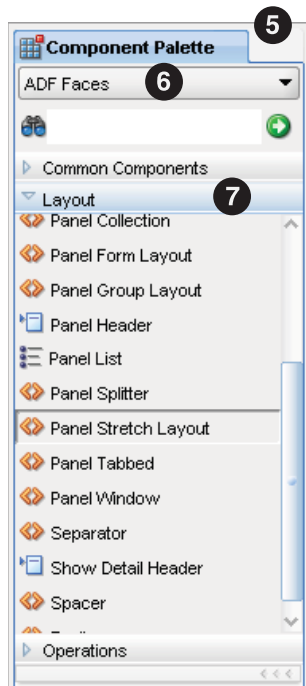
První kroky při tvorbě stránky aplikace spočívají v rozložení stránky do základních bloků. Navíc je nutné vytvořit základní grafickou úpravu stránky – odsazení pracovní plochy od okraje prohlížeče, zajištění automatického přizpůsobení velikosti pracovní plochy vůči velikosti okna HTML prohlížeče a umístění loga. V případě, že stránka obsahuje velký počet komponent, je nutné zajistit, aby stránka byla stále přehledná a nabízela rychlou orientaci. Osvědčeným řešením této situace je využití vzájemně se překrývajících ploch, které je možné snadno přepínat prostřednictvím záložek.



1. Vytvoření stránky aplikace

Nejprve otevřeme grafický editor souboru *faces-config.xml*, který najdeme v okně *Application Navigator*. ① Následně přesuneme položku *JSF Page* z okna *Component Palette* ② do plochy grafického editoru, kde se objeví ikona s názvem */untitled1.jspx*. ③ Pro vytvoření samotné stránky/souboru je nutné kliknout na ikonu a modifikovat hodnoty v dialogu *Create JSF JSP*. ④ Nastavíme atribut *File Name* na hodnotu *index.jspx* a v sekci *Page Implementation* vybereme volbu *Automatically Expose UI Components in a New Managed Bean*. Po ukončení dialogu se vytvoří soubory *Index.java* a *index.jspx*, které se současně otevrou v grafickém editoru.





2. Rozložení stránky do základních bloků

Z okna *Component Palette* 5 ze sady komponent *ADF Faces* 6 a skupiny *Layout* 7 vybereme komponentu *Panel Stretch Layout* a přesuneme ji do plochy grafického editoru. V okně *Property Inspector* 8 upravíme atributy právě vložené komponenty. Atribut *InlineStyle* upravíme na hodnotu *height:100%; width:100%; position:absolute*.

3. Vložení překrývajících se oblastí s navigací pomocí záložek

Nyní z okna *Component Palette* vybereme komponentu *Panel Tabbed* a přesuneme ji do prostřední oblasti označené textem *Facet center*. Jednotlivé záložky přidáme tak, že v okně *Structure* 9 vybereme položku reprezentující právě přidanou komponentu *Panel Tabbed* 10 a stiskem pravého tlačítka zobrazíme kontextové menu, kde vybereme položku *Insert inside af:panelTabbed* a její podpoložku *Show Detail Item*. 11 V okně *Property Inspector* upravíme název a identifikaci záložky změnou atributu *Text* na hodnotu *Master / Detail* a atributu *Id* na hodnotu *masterDetail*. Analogicky opakujeme sekvenci kroků pro druhou záložku, kterou nazveme *Graph* a atribut *Id* nastavíme na hodnotu *graph*.

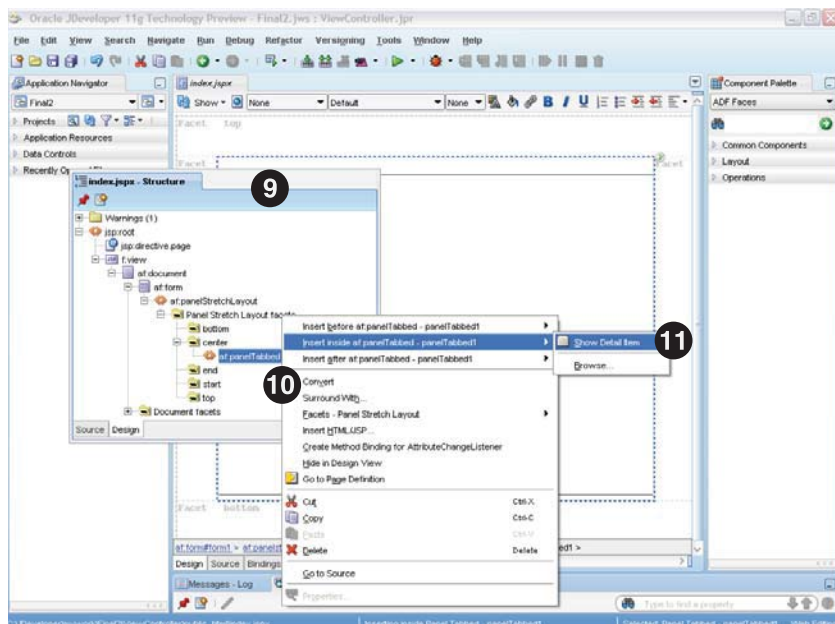
4. Grafické úpravy

Po aplikování předcházejících kroků se dostáváme do situace, kdy máme stránku správně logicky rozloženou a v případě grafického editoru JDeveloper i správně zobrazenou. V situaci, že aplikaci spustíme, zobrazí se v prohlížeči stránka se záložkami, které ale nemají žádný odstup od kraje okna prohlížeče. Pro docílení lepšího vzhledu je nutné umístit neviditelný obsah do jednotlivých oblastí *Panel Stretch Layout* a následně v grafickém editoru zvolit adekvátní šířku a výšku oblastí.

Pro vložení neviditelného obsahu využijeme komponentu *Spacer* (*ADF Faces/Layout*), kterou přesuneme postupně do oblastí *facet start* a *facet end*.

Dalším krokem je umístění loga ORACLE do horního pravého rohu stránky.

Nejprve umístíme komponentu *Panel Group Layout* do oblasti *facet top* a následně do ní vložíme komponentu *Image*, kde atribut *source* nastavíme na hodnotu */images/logo.gif*. Vložené komponentě *Panel Group Layout* upravíme nastavení atributů v okně *Property Inspector* na hodnoty (*Layout = vertical, Halign = right*).



Master / Detail

Nejčastěji využívaným návrhovým vzorem uživatelského rozhraní je tabulka a formulář, které jsou ve vztahu nadřazeného a porazeného, obecně známého pod pojmem Master–Detail. Uživatel může vybrat myší či klávesnicí řádek tabulky Departments a v tabulce Employees se mu automaticky zobrazí zaměstnanci tohoto oddělení. Analogicky může uživatel vybrat konkrétní řádek tabulky zaměstnanců a automaticky se zobrazí data ve formuláři. V ukázce využijeme vztahu databázových tabulek Departments a Employees.

Možnost změny pořadí a šířky sloupců

Filtrování

Řazení

Tabulka zaměstnanců, seznam se automaticky aktualizuje při změně výběru oddělení

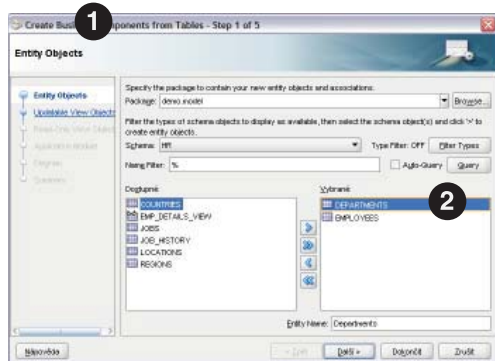
Editace názvu oddělení

Posuvník

Formulář pro editaci dat zaměstnance vybraného v tabulce zaměstnanců

V rámci ukázky si vystačíme s jednoduchou formou aplikační logiky, která bude pouze nabízet základní operace nad databázovými tabulkami (CRUD – Create, Read, Update, Delete). Pro ukázku jsme vybrali tabulky z databázového schématu „HR“ (Human Resources), které je standardně součástí instalace Oracle Database.

K implementaci CRUD operací využijeme jeden z dostupných O/R (Object – Relation) mapovacích nástrojů – ADF Business Components. Další ukázky budeme stavět nad informacemi o odděleních a jejich zaměstnancích, budeme tedy pracovat s entitami *Departments* a *Employees*.



Aplikační objekty vytvoříme pomocí průvodce *Create Business Components from Tables*. Postupně realizujeme kroky:

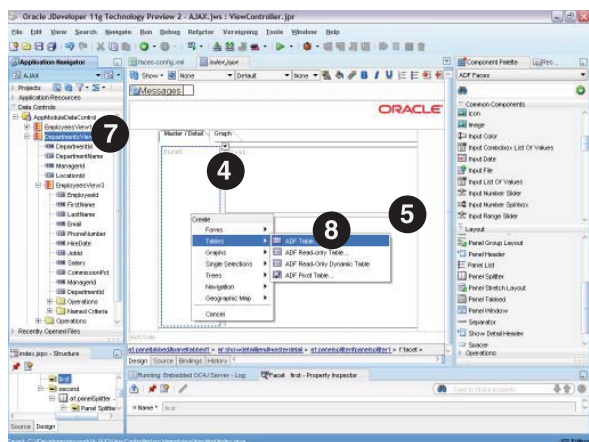
1. V okně *Application Navigator* vybereme projekt *Model* a v kontextovém menu zvolíme položku „New ...“
2. V zobrazeném dialogu vybereme kategorii *ADF Business Components* a položku *Business Components from Tables*.
3. Vytvoříme připojení k databázi, tj. vyplníme potřebné údaje k navázání spojení s databází (username = hr, password = hr).
4. Prvním krokem průvodce je výběr *Entit*, 1 kde vybereme tabulky *Departments* a *Employees*. 2
5. Ve výběru *Updatable View Object* vybereme obě nabízené položky. 3



V dalších krocích již necháme všechny volby ve výchozím nastavení.

1. Rozdělení pracovní plochy do 3 velikostí flexibilních částí

Do záložky „Master / Detail“ umístíme komponentu *Panel Splitter*, která rozdělí plochu na dvě vertikální oblasti. 4 Následně upravíme vlastnosti vložené komponenty tak, aby maximálně využila vymezenou oblast. V okně *Property Inspector* (záložka *Style*, záložka *Box*) nastavíme atributy *width* a *height* na hodnotu *100%*. Pravou oblast dále rozdělíme na dvě horizontální podoblasti, 5 tj. opět umístíme *Panel Splitter* do zmíněné oblasti. V *Property Inspector* zvolíme ve vlastnosti *Orientation* hodnotu *vertical*. 6



2. Vložení tabulky dovolující výběr a editaci názvu oddělení (Departments)

V okně *Application Navigator* (záložka *Data Controls*) vybereme položku *DepartmentsView1*, 7 stylem drag & drop jí přemístíme do levé oblasti stránky a ze zobrazeného menu vybereme položku *ADF Table...* ze sekce *Table*. 8

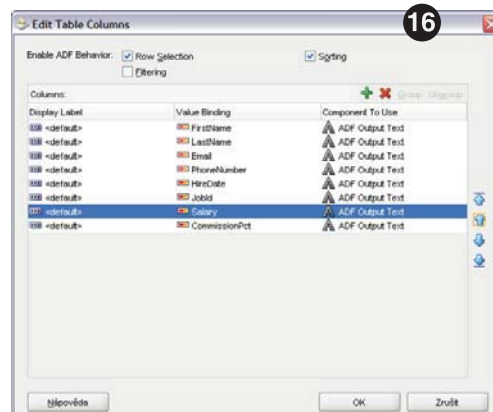
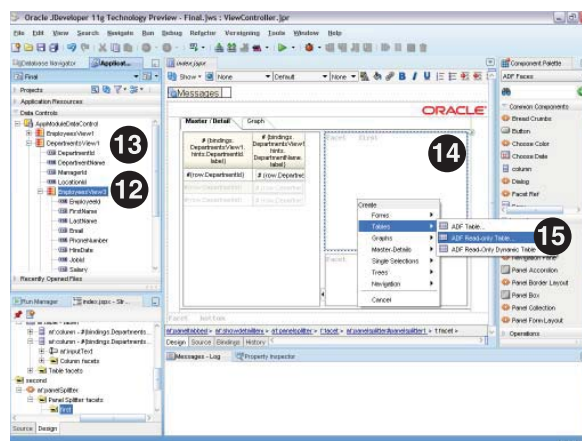
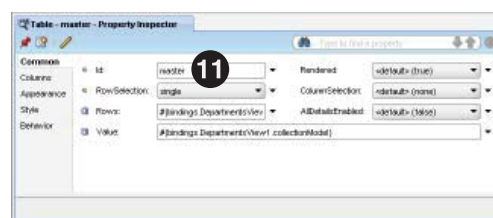


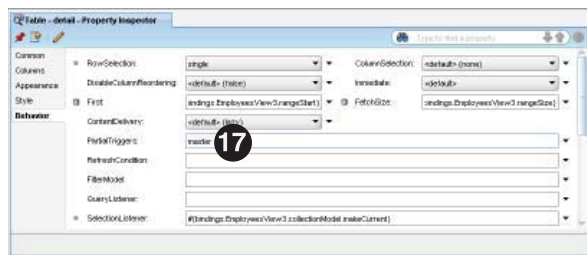
Po volbě se zobrazí dialog **9** s parametry vkládané tabulky, ve kterém zvolíme volby *Row selection, Sorting a Filtering*. **10** Následně modifikujeme konfiguraci sloupců tak, že odstraníme *ManagerId* a *LocationId* sloupce a u sloupce *DepartmentId* zvolíme komponentu *ADF Output Text*. Pro přehlednost budeme tabulku identifikovat názvem *master*, identifikaci provedeme výběrem komponenty *table* v okně *Structure* a následně modifikaci hodnoty *Id* v okně *Property Inspector*. **11**

3. Vložení tabulky dovolující výběr zaměstnance (Employees)

V okně *Application Navigator* vybereme položku *EmployeesView3* **12** (podřízený objekt objektu *DepartmentsView1*) **13** a stylem drag & drop ji přemístíme do pravé horní oblasti stránky **14** a ze zobrazeného menu vybereme položku *ADF Read-only Table...* **15**

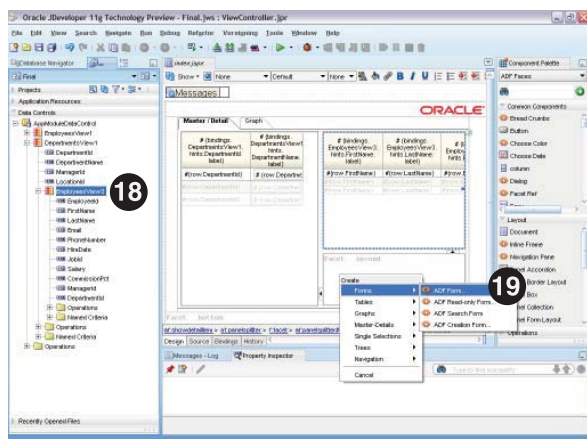
V zobrazeném dialogu provedeme úpravy tak, aby nastavení odpovídalo zobrazenému dialogu. **16** Podobně jako v případě předchozí tabulky provedeme změnu identifikace tabulky na hodnotu *detail*. Abychom zajistili, že při výběru záznamu v tabulce oddělení (*master*) dojde k překreslení tabulky zaměstnanců (*detail*), musíme definovat závislost mezi oběma prvky. Závislost definujeme uvedením hodnoty *master* **17** do atributu *PartialTriggers* komponenty *detail*.





4. Vložení formuláře pro editaci záznamu zaměstnance (Employees)

Do zbývajících prázdných oblastí stránky vložíme formulář dovolující editovat vybraný záznam tabulky *Employees*. V okně *Application Navigator* vybereme položku *EmployeesView3* **18** (podřízený objekt objektu *DepartmentsView1*) a stylem drag & drop jí přemístíme do pravé dolní oblasti stránky a ze zobrazeného menu vybereme položku *ADF Form...* **19**



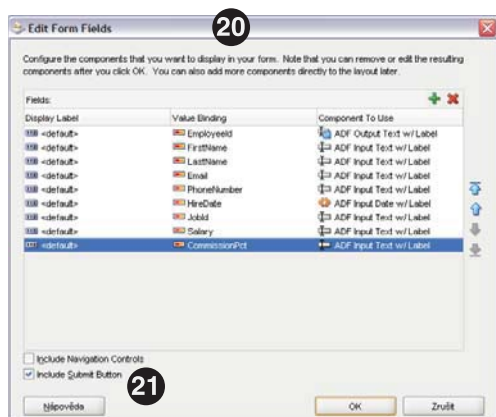
V zobrazeném dialogu **20** provedeme následující úpravu položek formuláře:

- Smazání položek (*DepartmentId*, *ManagerId*)
- Změna komponenty na *ADF Output Text w/Label* (*EmployeeId*)

Navíc zvolíme možnost *Include Submit Button*. **21**

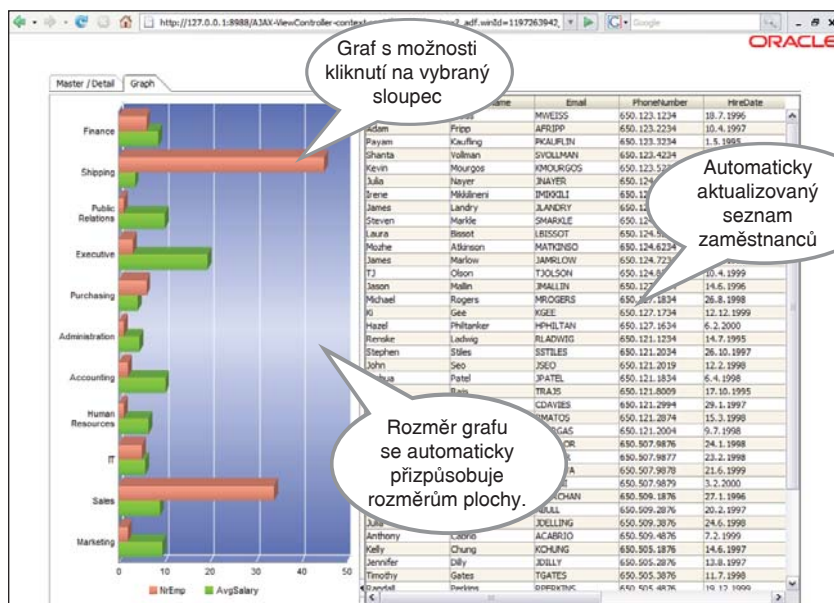
Opět důležitým krokem je deklarace závislosti formuláře na tabulce *detail*, tedy zajištění překreslení formuláře při změně výběru řádku v tabulce *detail*. Závislost definujeme uvedením hodnoty *detail* do atributu *PartialTriggers* nadřazené komponenty formuláře – *Panel Form Layout*.

Ve formuláři je umístěno tlačítko, jehož stisknutím dojde nejen k odeslání informací z formuláře, ale také k nežádoucímu opětovnému překreslení celé stránky aplikace. Toto chování změníme změnou atributu *PartialSubmit* komponenty *Command Button – Submit* na hodnotu *true*.



Interaktivní graf

„Jeden obrázek vydá za tisíce slov“ – je známé tvrzení, které dokládá význam vizualizace jako formy sdělení. Pro vizualizace numerických hodnot je v ADF Faces k dispozici celá řada předpřipravených grafů. V ukázce vytvoříme graf zobrazující jak počty zaměstnanců v jednotlivých odděleních a tak i průměrný plat v rámci oddělení. Navíc rozšíříme funkčnost grafu o interaktivitu, tedy o možnost kliknutí na sloupec grafu, po kterém se automaticky zobrazí seznam zaměstnanců vybraného oddělení.

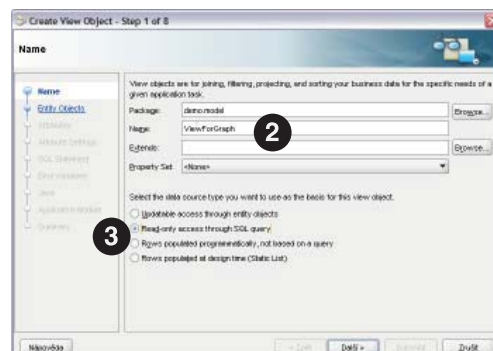
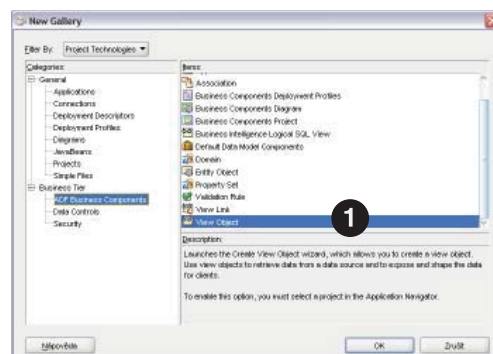


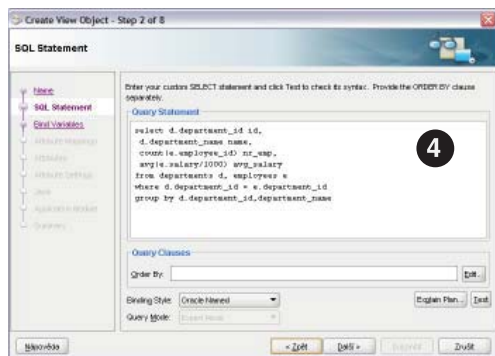
Zobrazený graf je založen na následujícím SQL dotazu, který vybere všechny údaje o oddělení včetně počtu zaměstnanců a průměrného platu (v tisících USD).

```
select d.department_id id,
       d.department_name name,
       count(e.employee_id) nr_emp,
       avg(e.salary/1000) avg_salary
from departments d, employees e
where d.department_id = e.department_id
group by d.department_id, department_name
```

Pro účely zobrazení grafu rozšíříme aplikační logiku ukázkové aplikace.

1. V okně **Application Navigator** vybereme projekt **Model** a v kontextovém menu zvolíme položku „New...“
2. V zobrazeném dialogu vybereme kategorii **ADF Business Components** a položku **View Object**.¹
3. Název objektu zvolíme **ViewForGraph** ² a variantu **Read-only access through SQL query**. ³





4. Do políčka *Query statement* 4 napíšeme výše uvedený SQL dotaz.

5. V kroku nazvaném *Application Module* zaškrtneme políčko *Application Module*. 5

Další kroky průvodce můžeme přeskočit.

Aplikační logika je připravena a přejdeme do tvorby uživatelského rozhraní.

1. Rozdělení pracovní plochy do dvou vertikálních částí

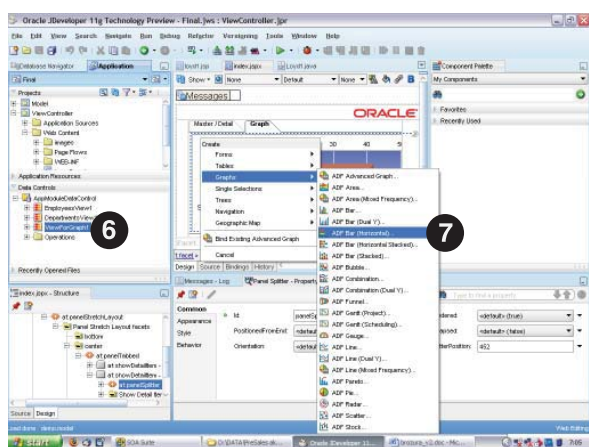
Do záložky „Graph“ umístíme komponentu *Panel Splitter*, která rozdělí plochu na dvě vertikální oblasti. Následně upravíme vlastnosti vložené komponenty tak, aby maximálně využila vymezenou oblast. V okně *Property Inspector* (záložka *Box*) nastavíme atributy *width* a *height* na hodnotu 100%.



2. Vložení a konfigurace grafu

V okně *Application Navigator* (záložka *Data Controls*) vybereme položku *ViewForGraph1*, 6 stylem drag & drop ji přemístíme do levé oblasti stránky a ze zobrazeného menu vybereme položku *ADF Bar (Horizontal)*... 7 ze sekce *Graphs*. V zobrazeném dialogu v sekci *Data Points* přidáme dvě položky (stisknutím tlačítka „+“). 8 U první položky zvolíme ve sloupci *Data Attribute* hodnotu *NrEmp* a u druhé položky zvolíme hodnotu *AvgSalary*. V sekci *Group By* přidáme jeden záznam a zvolíme hodnoty sloupců *Group Attribute* a *Group Label Attribute* na hodnoty *Id* respektive *Name*. Navíc máme možnost si ověřit,

jak vypadá námi nakonfigurovaný graf. Zvolíme záložku *Preview* 9 a po chvíli, kdy se JDeveloper přímo připojí k databázi, zobrazí se graf. Po vložení grafu do stránky budeme postupně modifikovat v okně *Property Inspector* položku *ImageFormat* na hodnotu *SVG*, položku *Id* na hodnotu *myGraph*, položku *3D Effect* na hodnotu *true*, položku *Style* na hodnotu *Confetti*, položku *DynamicResize* na hodnotu *DYNAMIC_SIZE* a položku *InlineStyle* na hodnotu *width:100%; height:100%;*.



3. Vložení tabulky zaměstnance (Employees)

V okně *Application Navigator* vybereme položku *EmployeesView3* (podřízený objekt objektu *DepartmentsView1*), stylem drag & drop ji přemístíme do pravé oblasti stránky a ze zobrazeného menu vybereme položku *ADF Read-only Table*... V zobrazeném dialogu provedeme úpravy tak, aby nastavení odpovídalo zobrazenému dialogu. 10

Důležitým krokem je deklarace závislosti vložené tabulky na grafu *myGraph*, kterým zajistíme překreslení tabulky pokud uživatel klikne na sloupec v grafu *myGraph*. Závislost definujeme v okně *Property Inspector* uvedením hodnoty *myGraph* do atributu *PartialTriggers*. Tabulka s výpisem zaměstnanců daného oddělení je založena na identickém datovém modelu (Iterator) na jakém jsou založeny tabulky v záložce „Master / Detail“. Tento fakt si jednoduše potvrdíme při běhu aplikace tím, že v záložce „Master / Detail“ například vybereme v tabulce oddělení *Finance* a při přepnutí na záložku „Graph“ se zobrazí právě 6 zaměstnanců oddělení *Finance*. Naopak nyní v grafu například klikneme na sloupec u oddělení *Administration* a při návratu na záložku „Master / Detail“ se objeví v tabulce oddělení zvýrazněný řádek *Administration*.

4. Provázání grafu a tabulky zaměstnanců

Narozdíl od tabulek Master / Detail, kde koncept ADF automaticky zajišťuje synchronizaci tabulek oddělení a zaměstnanců, musíme v tomto případě synchronizaci zajistit my. Synchronizaci zajistíme díky vložení metody *myClickListener*, která je automaticky volána při kliknutí na sloupce grafu a obsahuje Java kód, který dle zvoleného sloupce upraví hodnotu iterátoru *DepartmentsView1*.

Vložení metody *myClickListener* provedeme v okně *Property Inspector* vložení hodnoty *myClickListener* do atributu *ClickListener* komponenty *myGraph*. Zadááním hodnoty *myClickListener* se automaticky přidá metoda *myClickListener* (*ClickEvent clickEvent*). ⑪ Vygenerovaný Java kód nahradíme následujícím Java kódem, včetně atributu *bindings*.

```
private BindingContainer bindings;

public BindingContainer getBindings() {
    return this.bindings;
}

public void setBindings(BindingContainer bindings) {
    this.bindings = bindings;
}

public void myClickListener(ClickEvent clickEvent) {
    DataComponentHandle dch = (DataComponentHandle)clickEvent.getComponentHandle();

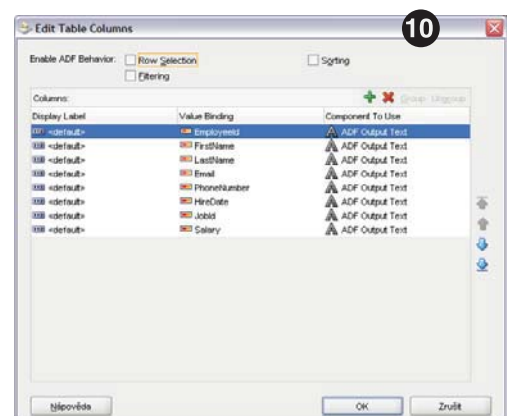
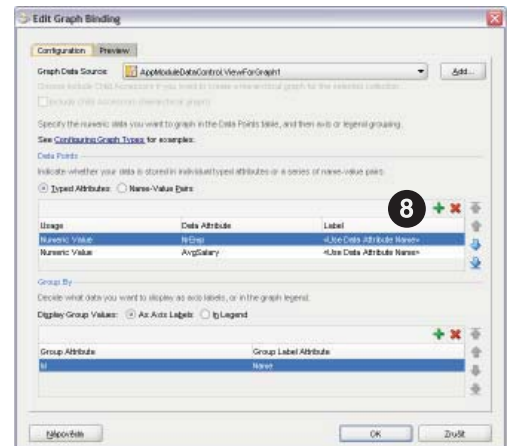
    UIGraph graph = (UIGraph)clickEvent.getSource();
    int iColNum = ((DataComponentHandle)dch).getColumn();
    int iRowNum = ((DataComponentHandle)dch).getRow();

    String id = null;
    try {
        id =
            (String)graph.getGraphDataModel().getDataAccess().getMemberMetadata(DataDirector.COLUMN_EDGE,
                0,
                iColNum,
                MetadataMap.METADATA_VALUE);
    } catch (Exception e) {
        e.printStackTrace();
    }

    DCIteratorBinding pr_dcib = (DCIteratorBinding)bindings.get("DepartmentsView1Iterator");
    pr_dcib.setCurrentRowWithKeyValue(id);
}
```

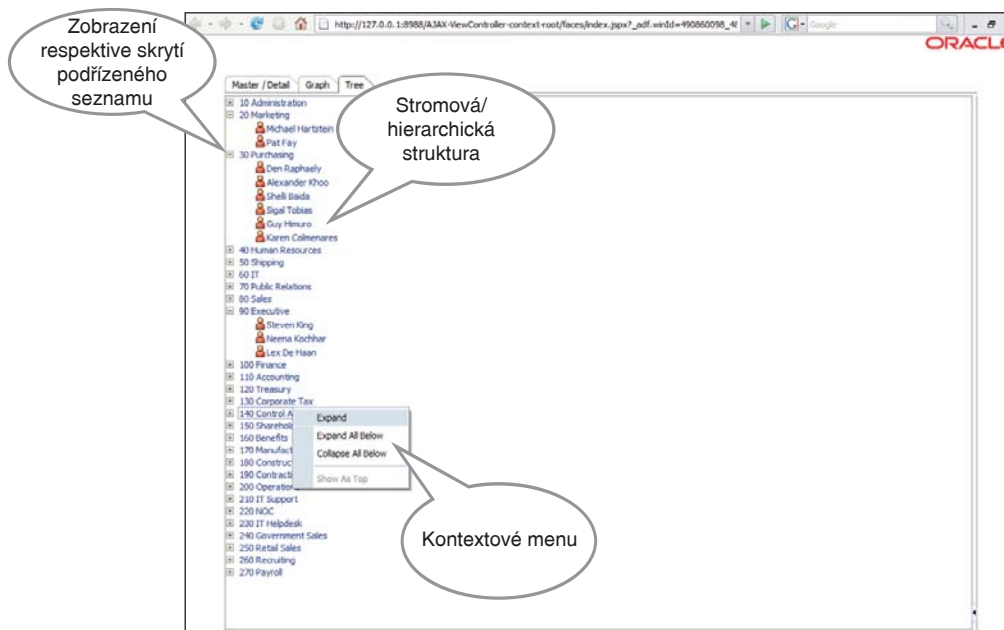
Do souboru *adfc-config.xml* je nutné umístit následující XML fragment – tag *<managed-property>*. XML text umístíme do tagu *<managed-bean>*.

```
<managed-bean>
  <managed-bean-name>backing_index</managed-bean-name>
  <managed-bean-class>demo.view.backing.Index</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <!--oracle-jdev-comment:managed-bean-jsp-link: lindex.jspx-->
  <managed-property>
    <property-name>bindings</property-name>
    <value>#{bindings}</value>
  </managed-property>
</managed-bean>
```



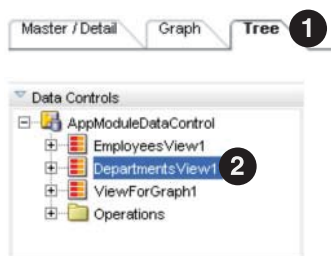
Hierarchie

Zajímavým ovládacím prvkem uživatelského rozhraní je grafické zobrazení hierarchické struktury, známé například z adresářové struktury souborového systému. Pro tyto účely se běžně používá zobrazení ve formě stromové struktury. V této kapitole ukážeme implementaci ovládacího prvku typu „tree“, který využijeme pro zobrazení dvouúrovňového seznamu. První úroveň bude tvořit seznam oddělení (departments) a druhou úroveň bude tvořit seznam příslušných zaměstnanců (employees). Koncový uživatel může kliknutím na symboly „+“ respektive „-“ zobrazit nebo naopak skrýt seznam zaměstnanců příslušného oddělení.



zadáni

postup



1. Přidání další navigační záložky Tree ①

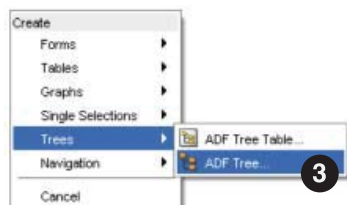
Záložky přidáme tak, že v okně **Structure** vybereme položku reprezentující komponentu **Panel Tabbed** a stiskem pravého tlačítka zobrazíme kontextové menu, kde vybereme položku **Insert inside af:panelTabbed** a její podpoložku **Show Detail Item**. V okně **Property Inspector** upravíme název a identifikaci záložky změnou atributu **Text** na hodnotu **Tree** a atributu **Id** na hodnotu **tree**.

2. Rozdělení stránky do dvou částí

Do záložky „Tree“ umístíme komponentu **Panel Splitter**, která rozdělí plochu na dvě vertikální oblasti. Následně upravíme vlastnosti vložené komponenty tak, aby maximálně využila vymezenou oblast. V okně **Property Inspector** (záložka **Box**) nastavíme atributy **width** a **height** na hodnotu **100%**.

3. Ovládací prvek typu strom

Do levé části právě rozdělené stránky vložíme ovládací prvek typu strom. V okně **Data Controls** vybereme položku **DepartmentsView1** ② a přesuneme ji do levé části stránky. Ze zobrazené nabídky zvolíme položku **ADF Tree Table...** ③ Po výběru se zobrazí dialog, kde postupně konfiguruje dvě úrovně stromu.

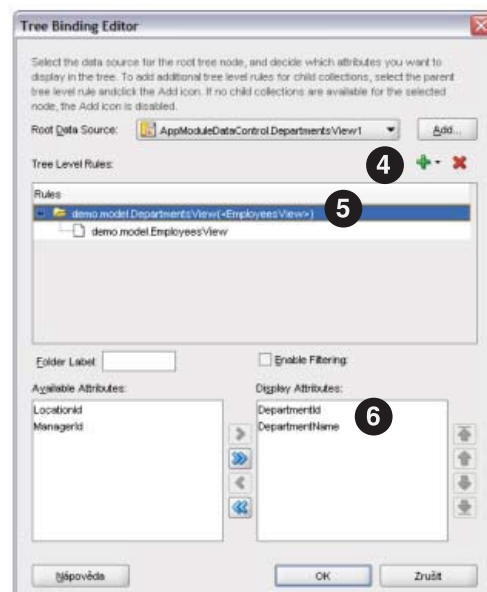


Jednotlivé úrovně vkládáme pomocí tlačítka „+“, ④ kde se nejprve objeví text *AddRule* a následně po vložení a výběru *DepartmentsView* ⑤ se objeví text *EmployeesView*. Po vložení dvou úrovní nakonfiguruje atributy, které se zobrazí v jednotlivých úrovních stromu. Pro první úroveň (departments) zvolíme atributy *DepartmentId* a *DepartmentName*, ⑥ pro druhou úroveň (employees) zvolíme atributy *FirstName* a *LastName*. ⑦

4. Přidání grafické ikony k položce zaměstnance

Grafické zobrazení stromové struktury vylepšíme přidáním ikony do druhé úrovně stromu. V tomto případě přímo upravíme zdrojový kód stránky. Před zobrazovaný text položky stromu vložíme tag `<af:image>`, který podmíněně zobrazí ikonu symbolizující zaměstnance. Upravený kód je zobrazen níže. Podmínka rozhodující o zobrazení či nezobrazení ikony je založena na atributu *depth*, který udává hloubku právě zobrazované položky ve stromě (pro nejvyšší úroveň se ikona nezobrazuje).

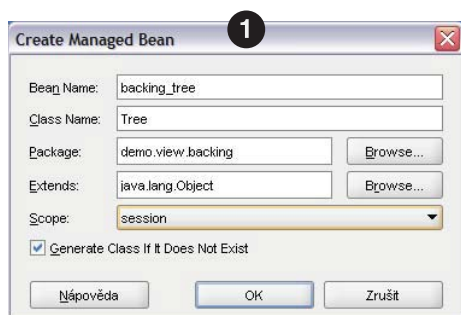
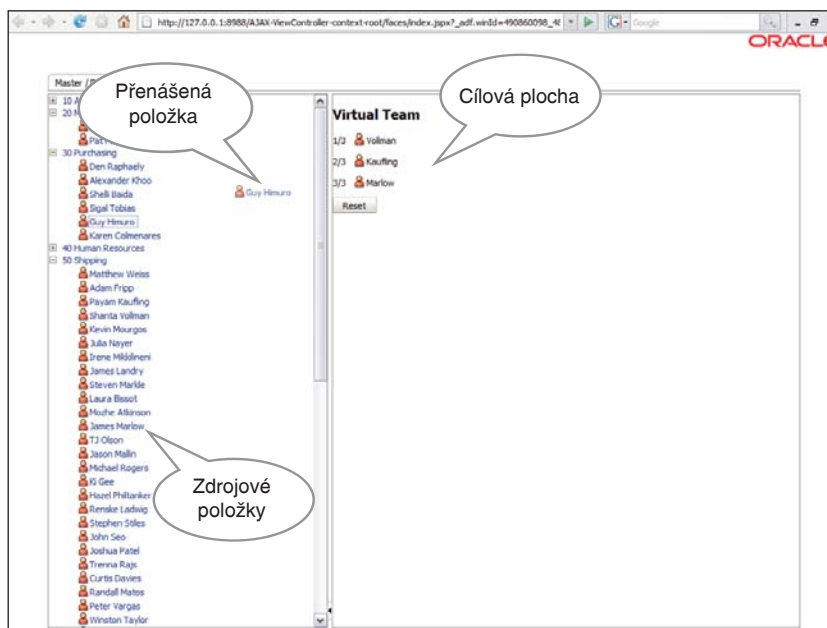
```
<af:tree value="#{bindings.DepartmentsView1.treeModel}"
  var="node" binding="#{backing_index.treel}"
  id="treel">
  <f:facet name="nodeStamp">
    <af:panelGroupLayout>
      <af:image source="#{backing_index.treel.depth==0 ? null:'/images/guy.gif'}"
        inlineStyle="vertical-align:bottom;"/>
      <af:outputText value="#{node}"
        id="outputText2"/>
    </af:panelGroupLayout>
  </f:facet>
</af:tree>
```



Drag & Drop

Velmi efektivním způsobem práce v aplikaci je využití techniky „drag & drop“. Jedná se o možnost přesunu určité položky pomocí myši z jednoho místa na jiné. S touto činností je pak asociována funkce aplikační logiky, například přemístění souboru mezi adresáři, úprava pořadí sloupců tabulky, atp.

V našem případě budeme vytvářet „virtuální tým“ – tedy skupinu zaměstnanců složenou z různých oddělení. Ve výsledné aplikaci bude možné přemísťovat jednotlivé zaměstnance zobrazené ve stromové struktuře do plochy zobrazující seznam členů virtuálního týmu.



1. Přidání nové asociované Java třídy

V případě přidání funkčnosti „drag & drop“ do stránky *index.jspx* již není dále možné mít pouze jednu asociovanou Java třídu. Příčina spočívá v současné potřebě dvou rozsahů platnosti Java třídy – rozsah platnosti *request* a rozsah platnosti *session*. Rozsah *session* je nutný z důvodu uchování seznamu *virtualTeam* mezi jednotlivými uživatelskými dotazy (přesuny položek) a rozsah *request* je vyžadován pro účely implementace interaktivních grafů (inicializace hodnoty atributu *binding*). Původní asociované Java třídy *index.java* ponecháme rozsah platnosti *request*, druhé asociované třídy *tree.java* nastavíme rozsah platnosti na hodnotu *session*.

Asociovanou Java třídu *tree.java* vytvoříme aplikací následujícího postupu:

1. otevřeme soubor *faces-config.xml*,
2. zvolíme záložku *Overview*,
3. zvolíme kategorie *Managed Beans*,
4. stiskneme tlačítko *New*
5. a vyplníme položky dialogu 1 podle zobrazeného vzoru.

Do nově vytvořené Java třídy *tree.java* vložíme následující kód.

```
public class Tree {
    private UIXIterator iterator1;
    private List<String> virtualTeam;
    public Tree() {
        virtualTeam = new ArrayList<String>();
    }
    public void setIterator1(UIXIterator iterator1) {
        this.iterator1 = iterator1;
    }
    public UIXIterator getIterator1() {
        return iterator1;
    }
    public String reset_action(){
        virtualTeam.clear();
        return null;
    }
    public DnDAction handleDrop(DropEvent dropEvent) {
        try {
            DataFlavor<String> df = DataFlavor.getDataFlavor(String.class);
            String droppedValue = dropEvent.getTransferable().getData(df);
            if (dropEvent == null) {
                return DnDAction.NONE;
            } else {
                virtualTeam.add(droppedValue);
            }
            AdfFacesContext.getCurrentInstance().addPartialTarget(getIterator1());
            return DnDAction.COPY;
        } catch (Exception ex) {
            return DnDAction.NONE;
        }
    }
    public void setVirtualTeam(List<String> newvirtualTeam) {
        this.virtualTeam = newvirtualTeam;
    }
    public List<String> getVirtualTeam() {
        return virtualTeam;
    }
}
```

2. Zdrojové položky pro funkčnost „drag & drop“

V předchozí kapitole jsme vytvořili ovládací prvek typu strom. Tento prvek využijeme pro vytvoření zdrojových položek. Do kódu stránky, do tagu `<af:panelGroupLayout>`, vložíme dva tagy – `<af:clientAttribute>` a `<af:attributeDragSource>`, které povolují funkčnost „drag & drop“ a definují data, která jsou asociována s konkrétním přesunem. Výsledný kód má následující tvar a vyplývá z něho, že asociovaná data jsou tvořena příjmením zaměstnance (lastName) a to pouze v případě, že se jedná o položky stromu v druhé úrovni. Není tedy možné přenášet oddělení, ale pouze zaměstnance.

```
<af:tree value="#(bindings.DepartmentsView1.treeModel)"
    var="node" binding="#(backing_index.treel)"
    id="treel">
    <f:facet name="nodeStamp">
        <af:panelGroupLayout>
            <af:image source="#(backing_index.treel.depth==0 ? null:'/images/guy.gif')"
                inlineStyle="vertical-align:bottom;"/>
            <af:outputText value="#(node)"
                id="outputText2"/>
            <af:clientAttribute name="textik" value="#(backing_index.treel.depth==1 ? node.LastName : null)"/>
            <af:attributeDragSource attribute="textik"/>
        </af:panelGroupLayout>
    </f:facet>
</af:tree>
```

3. Cílový seznam pro funkčnost „drag & drop“

Jednotlivé zaměstnance bude možné přenášet do pravé části stránky, kde bude zobrazen seznam již přenesených zaměstnanců. Do pravé oblasti stránky nejprve umístíme nadpis „Virtual Team“ (tag <H2>) a následně umístíme výpis seznamu přenesených zaměstnanců (tag <af:iterator>) a tlačítko pro smazání seznamu zaměstnanců (tag <af:commandButton>). Konkrétní kód získáte postupnou volbou ADF Faces komponent, nebo kopírováním následujícího kódu.

```
<f:facet name="second">
  <af:panelGroupLayout partialTriggers="clear">
    <h2>Virtual Team</h2>
    <af:iterator var="row" varStatus="stat" binding="#{backing_tree.iterator1}"
      value="#{backing_tree.virtualTeam}" id="iterator1">
      <p>
        <af:outputText value="#{stat.count}/#{stat.model.rowCount} &nbsp;&nbsp;&nbsp;"/>
        4 <af:image source="/images/guy.gif" inlineStyle="vertical-align:bottom;"/>
        <af:outputText value="&nbsp;&nbsp;&nbsp;#{row}"/>
      </p>
    </af:iterator>
    <af:commandButton text="Reset" id="clear"
      partialSubmit="true" action="#{backing_tree.reset_action}"/> 6
    <af:dropTarget droplistener="#{backing_tree.handleDrop}"
      <af:dataFlavor flavorClass="java.lang.String"/> 5
    </af:dropTarget>
  </af:panelGroupLayout>
</f:facet>
```

Poznámky ke zdrojovému kódu:

- Důležitým atributem tagu <af:iterator> je atribut *value*, který má hodnotu *backing_tree.virtualTeam* a odkazuje tak na atribut resp. metody asociované Java třídy *tree.java*. Podřízené tagy (<p>, <af:image>, <af:outputText>) 4 již definují zobrazení jedné položky seznamu.
- Pro funkčnost „drag & drop“ je nepostradatelný tag <af:dropTarget>, který definuje místo, kam je možné přesouvat položky. Přesunem položky uživatel požaduje vykonání určité akce, která je implementována metodou *handleDrop()* 5 (viz. hodnota atributu *dropListener*).
- Tag <af:commandButton> obsahuje odkaz na metodu *reset_action()*, 6 kde je implementována reakce na stisk tlačítka – smazání seznamu *virtualTeam*.