

Budowa warstwy biznesowej opartej na komponentach bezstanowych i sesyjnych typu EE. Utrwalanie danych za pomocą technologii JPA

<https://docs.oracle.com/javaee/7/JEETT.pdf>

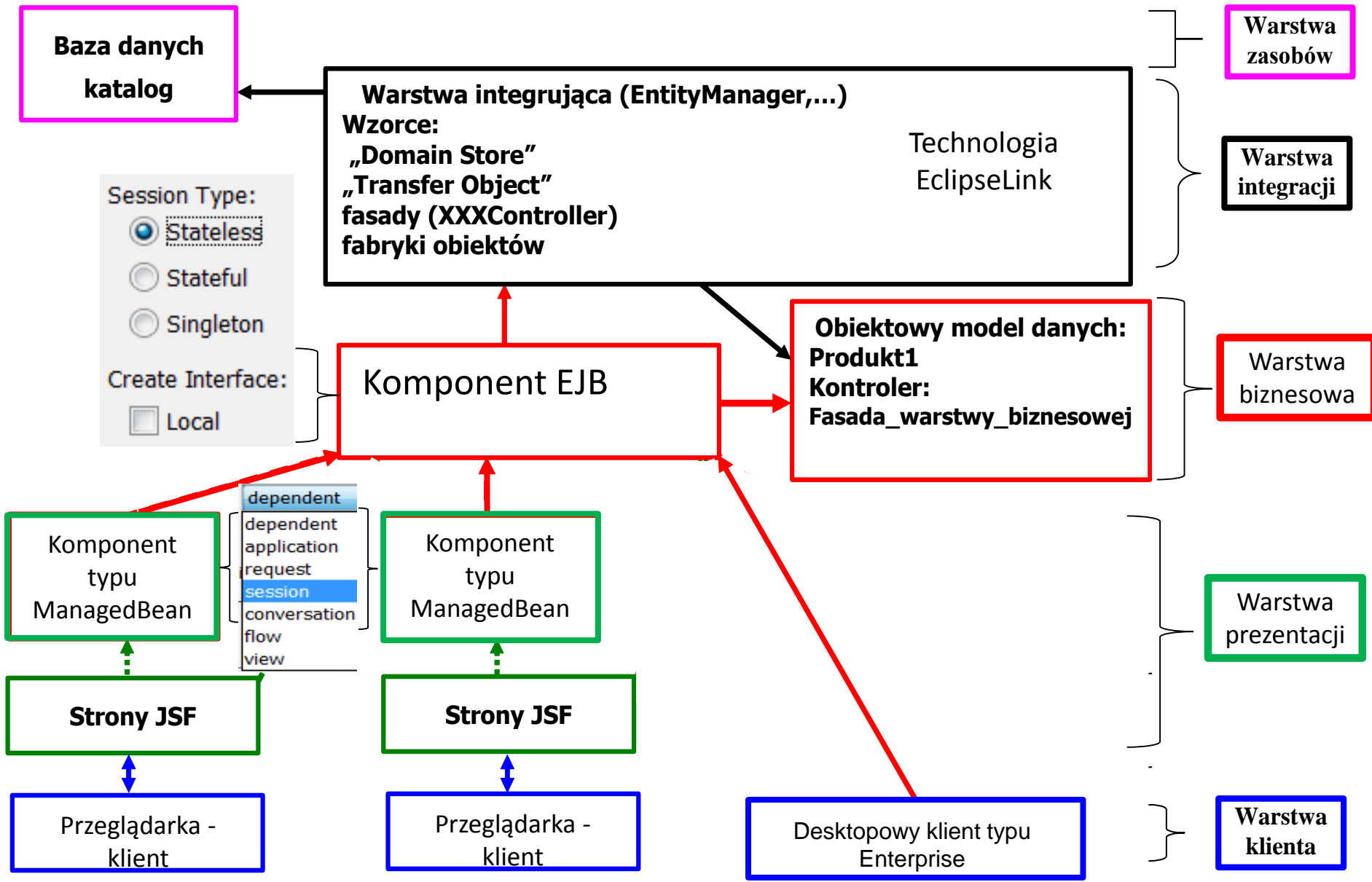
Programowanie komponentowe 6

1. Aplikacja wielowarstwowa Java EE oparta na komponentach EJB typu Session Bean – rodzaj Stateless lub Singleton

Warstwa internetowa oparta na komponentach Managed Bean o czasie życia **SessionScope**

Architektura aplikacji pięciowarstwowej -Java EE 7.0

JavaServer Faces

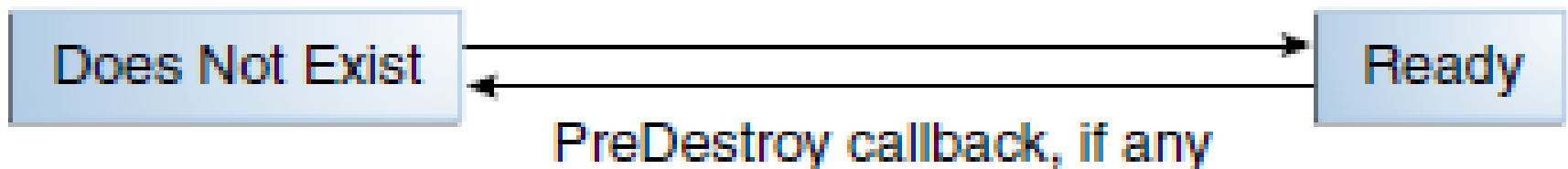


Cykl życia bezstanowych (**Stateless**) komponentów typu **Session Bean** – wykłady: 2, 4

Ponieważ bezstanowy komponent typu **Session Bean** nigdy nie przechodzi w stan pasywny, cykl życia składa się tylko z dwóch stanów: **nonexistent and ready** podczas wywoływania metod przez warstwę klienta.

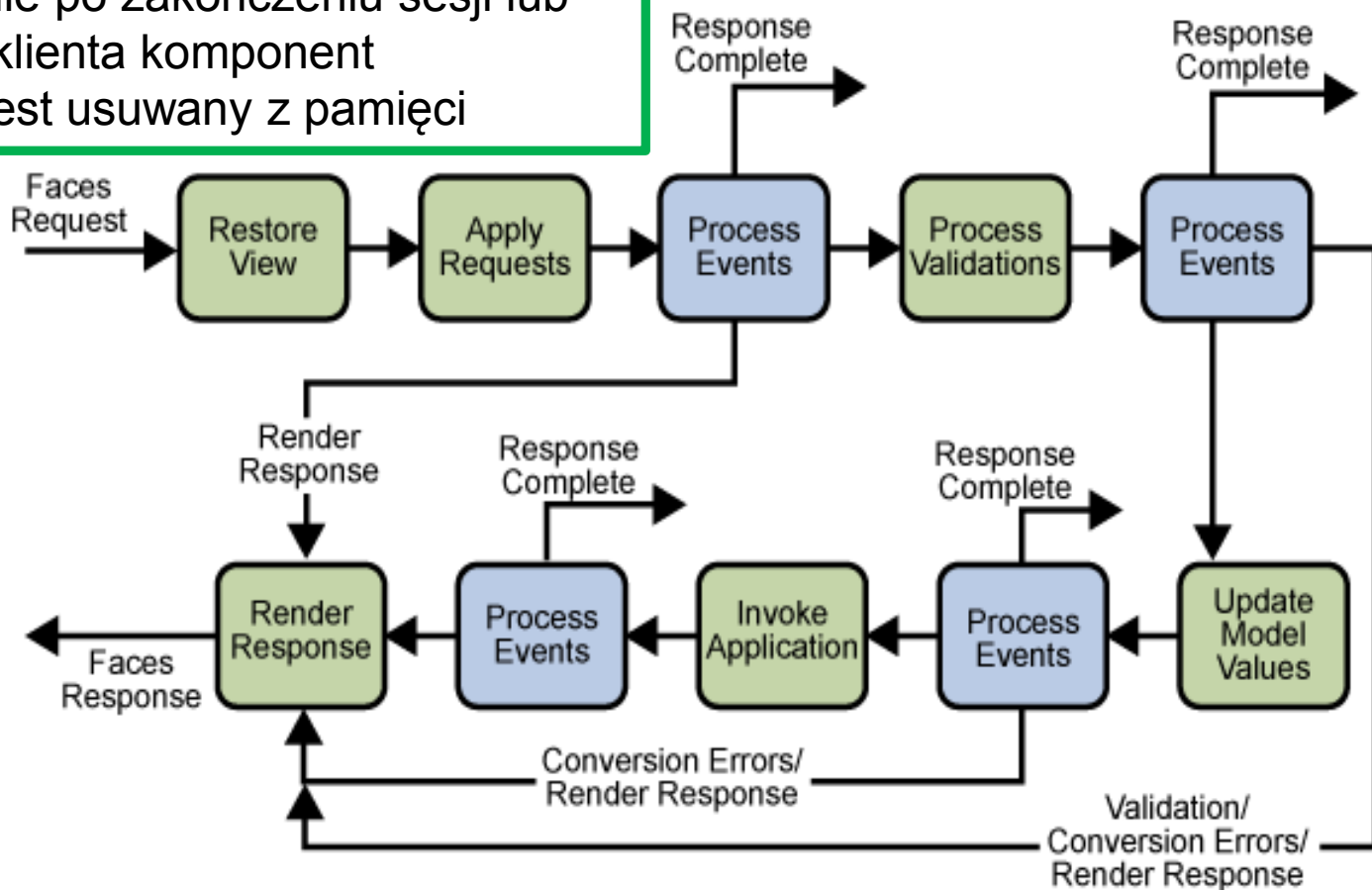
Zwykle istnieje wiele równoważnych instancji komponentu typu **Session obsługujących aplikacje klienckie**, zarządzanych przez kontener EJB – poprawia to **skalowalność aplikacji obsługujących wiele aplikacji klienckich** (lepiej niż komponenty typu **Stateful**).

- ① **Dependency injection, if any**
- ② **PostConstruct callback, if any**



Standard cyklu życia „Request-Response” dla JavaServer Faces

SessionScope: podczas uruchomienia instancji aplikacji klienta tworzony jest nowy komponent typu **Managed_produk**t – wszystkie jego dane są zaktualizowane podczas fazy Invoke Application. Po zakończeniu fazy Render Response komponent pozostaje w pamięci. Jeśli nie wystąpią błędy, jedynie po zakończeniu sesji lub zamknięciu aplikacji klienta komponent **Managed_produk**t jest usuwany z pamięci



1.1. Dodanie stronicowania zawartości komponentu **dataTable** na stronie **lista_produkow.xhtml**.

Kod dotyczący stronicowania oparty na kodzie wygenerowanym podczas tworzenia stron JSF na podstawie klas typu Entity (pakiet jsf.util) –

Dodatek do wykładu 2 (str. 32-85):

http://zofia.kruckiewicz.staff.iiar.pwr.wroc.pl/wyklady/ti_/TINT_2.pdf

Stronicowanie komponentu typu **dataTable**

```
<h:outputText value="#{managed_produk.t.pagination.pageFirstItem + 1}  
..#{managed_produk.t.pagination.pageLastItem + 1}  
/#{managed_produk.t.pagination.itemsCount}"/>
```

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks

Lista produktow

localhost:8080/SklepPl

Dodaj produkt
Lista produktow

4 ..6 /7 Poprzedni 3 Nastepny 3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
4	Produkt4	440 zł	50 %	wtorek, 28-03-2017	220 zł
5	Produkt5	550 zł	10 %	wtorek, 28-03-2017	495 zł
6	Produkt6	660 zł	10 %	wtorek, 28-03-2017	594 zł

Powrot

```
<h:commandLink action="#{managed_produk.t.next}"  
value="#{bundle['lista_produk.t.nastepny']}"  
#{managed_produk.t.pagination.pageSize}"  
rendered="#{managed_produk.t.pagination.hasNextPage}"/>
```

```
<h:commandLink action="#{managed_produk.t.previous}"  
value="#{bundle['lista_produk.t.poprzedni']}"  
#{managed_produk.t.pagination.pageSize}"  
rendered="#{managed_produk.t.pagination.hasPreviousPage}"/>
```

Dodanie stronicowania stron – fragment pliku `lista_produkow.xhtml` z dodanym kodem znaczników odpowiedzialnym za stronicowanie (zaznaczony kolorem czerwonym – przed znacznikiem `dataTable`) – pierwszy znacznik `commandLink` służy do stronicowania „wstecz” a drugi do stronicowania „do przodu”

```
<ui:define name="content">
  <h:form styleClass="jsfcrud_list_form">
    <h:panelGroup id="messagePanel" layout="block">
      <h:messages errorStyle="color: red" infoStyle="color: green" layout="table"/>
    </h:panelGroup>
    <h:outputText escape="false" value="#{bundle['lista_produkow.pusta']}"
      rendered="#{managed_produkow.pagination.itemsCount == 0}"/>
    <h:panelGroup rendered="#{managed_produkow.pagination.itemsCount > 0}">
      <h:outputText value="#{managed_produkow.pagination.pageFirstItem + 1}
        ..#{managed_produkow.pagination.pageLastItem + 1}
        /#{managed_produkow.pagination.itemsCount}"/>&nbsp;
      <h:commandLink
        action="#{managed_produkow.previous}"
        value="#{bundle['lista_produkow.poprzedni']} #{managed_produkow.pagination.pageSize}"
        rendered="#{managed_produkow.pagination.hasPreviousPage}"/>&nbsp;
      <h:commandLink
        action="#{managed_produkow.next}"
        value="#{bundle['lista_produkow.nastepny']} #{managed_produkow.pagination.pageSize}"
        rendered="#{managed_produkow.pagination.hasNextPage}"/>&nbsp;
    </h:panelGroup>
  </h:form>
</ui:define>
```


Zmiana czasu życia obiektu typu **Managed_produk**t do czasu trwania sesji za pomocą adnotacji **@SessionScoped**

```
package warstwa_internetowa;
```

```
/* importy */
```

```
@Named(value = "managed_produk")
```

```
@SessionScoped
```

```
public class Managed_produk implements ActionListener, Serializable {
```

```
public DataModel getItems() {
```

```
    if (items == null || fasada.isStan()) {
```

```
        items = getPagination().createPageDataModel(); }
```

```
    return items; }
```

W przypadku, gdy komponent typu Managed_produk ma czas życia sesji, należy w sposób algorytmiczny aktualizować dane produktów za pomocą metody **getItems** do wyświetlania, pobierając z miejsca przechowywania, czyli z serwera, gdy:

- badanie **items==null** wynika z konieczności pobrania danych do widoku komponentu dataTable, gdy dodano nowe dane produktów z klienckiej aplikacji internetowej () – metoda **recreateModel** ustawia wartość items na null,
- a warunek **fasada.isStan()** wynika z badania, czy z aplikacji desktopowej lub innej internetowej nie wstawiono nowych danych.

Kod, który należy dodać i zmodyfikować w klasie **Managed_produk**t.

```
public void dodaj_produk() {  
    fasada.utworz_produk(dto);  
    dane_produk();  
    recreateModel();  
    getPagination().nextPage();  
}
```

Konieczność modyfikacji zawartości wyświetlanej strony komponentu typu dataTable po wstawieniu nowego elementu:

- 1) **recreateModel** – ustawienie stanu do utworzenia nowego modelu strony
- 2) **getPagination().nextPage()** – sprawdzenie, czy nie należy utworzyć kolejnej strony tabeli

```
public void dane_produk() {  
    stan = 1;  
    produkt_dto = fasada.dane_produk();  
    if (produkt_dto == null) {  
        produkt_dto = new Produkt_dto();  
        stan = 0; }  
}
```

private PaginationHelper pagination; //nowy atrybut klasy **Managed_produk**

```
public PaginationHelper getPagination() {  
    if (pagination == null) {  
        pagination = new PaginationHelper(3) {
```

```
        @Override
```

```
        public int getItemCount() {  
            return getFasada().count();  
        }
```

```
        @Override
```

```
        public DataModel createPageDataModel() {  
            int[] range = {getPageFirstItem(), getPageLastItem() + 1};  
            return new ListDataModel(getFasada().findRange(range));  
        }
```

```
    };
```

```
    }  
    return pagination;
```

```
}
```

Ustalenie wartości atrybutu **pageSize**, w obiekcie klasy dziedziczącej po klasie abstrakcyjnej **PaginationHelper**, obsługującej stronicowanie komponentu **dataTable** (definicję klasy podano na kolejnych slajdach) z zaimplementowanymi metodami **getItemCount** oraz **createPageDataModel**

```
private void recreateModel() {  
    items = null;  
}
```

Metoda usuwająca zawartość bieżącej strony – umożliwia metodzie **getItems** utworzenie zawartości wybranej strony w sposób algorytmiczny

```
public String next() {  
    getPagination().nextPage();  
    recreateModel();  
    return "lista_produktow";  
}
```

Metoda obsługująca wybór następnej strony (str. 7-8)

```
public String previous() {  
    getPagination().previousPage();  
    recreateModel();  
    return "lista_produktow";  
}
```

Metoda obsługująca wybór poprzedniej strony (str. 7-8)

Do pakietu **pomoc** należy dodać klasę **PaginationHelper**, typu **abstract** zawierającą następujący kod, obsługujący stronicowanie

```
package pomoc;
```

```
import javax.faces.application.FacesMessage;
```

```
import javax.faces.context.FacesContext;
```

```
public abstract class PaginationHelper {
```

```
    private int pageSize;
```

```
    private int page;
```

```
    public PaginationHelper(int pageSize) {
```

```
        this.pageSize = pageSize;
```

```
    }
```

```
    public abstract int getItemCount();
```

```
    public abstract DataModel createPageDataModel();
```

```
    public int getPageFirstItem() {
```

```
        return page * pageSize; }  
}
```

pageSize - rozmiar strony
– czyli liczba wierszy
komponentu **dataTable**
page – numer strony

pageSize*page – liczba
pozycji przypisanych do
wszystkich stron, a
jednocześnie numer
danej wyświetlanej jako
pierwsza pozycja na
ostatniej stronie

```
public int getPageLastItem() {  
    int i = getPageFirstItem() + pageSize - 1;  
    int count = getItemsCount() - 1;  
    if (i > count) {  
        i = count; }  
    if (i < 0) {i = 0; }  
    return i;  
}
```

Indeks ostatniej pozycji, czyli numer ostatniej danej, wyświetlany na ostatniej stronie

```
public boolean hasNextPage() {  
    return (page + 1) * pageSize + 1 <= getItemsCount();  
}
```

Sprawdzenie, czyli liczba danych wymaga utworzenia następnej strony

```
public void nextPage() {  
    if (hasNextPage()) {  
        page++;  
    }  
}
```

Numer ostatniej strony

```
public boolean isHasPreviousPage() {  
    return page > 0;  
}
```

Sprawdzenie, czy istnieje poprzednia strona (pierwsza strona ma numer 0)

```
public void previousPage() {  
    if (isHasPreviousPage()) {  
        page--;  
    }  
}
```

Pobranie numeru poprzedniej strony (str 7-8)

```
public int getPageSize() {  
    return pageSize;  
}
```

Pobranie rozmiaru strony

```
public void setPage() {  
    this.page = getItemCount()/pageSize;  
}
```

Aktualizacja liczby stron

Kod, który należy dodać do klasy **Fasada_warstwy_biznesowej**

```
public boolean isStan() {  
    return stan; }  
public void setStan(boolean stan) {  
    this.stan = stan; }  
public int count() {  
    return produkty.size(); }  
public ArrayList<Produkt_dto> findRange(int[] range) {  
    ArrayList<Produkt_dto> pom = new ArrayList();  
    if (getProdukty().isEmpty()) {  
        stan = false;  
        return pom;  
    }  
    for (int i = range[0]; i < range[1]; i++)  
        pom.add(produkt_transfer(getProdukty().get(i)));  
    return pom;  
}
```

Pobranie informacji, czy dodano nowe obiekty typu **Produkt1** do kolekcji produkty

Pobranie podzbioru danych potrzebnych do wyświetlenia na stronie za pomocą metody **findRange**. Tablica **range** zawiera dwa elementy: pierwszy zawiera numer pierwszego elementu, drugi element zawiera numer ostatniego elementu z kolekcji **produkty**, które wyznaczają podzbiór danych pobieranych do wyświetlenia na stronie. Pobrane elementy z kolekcji produktu są przekształcone na obiekty transferowe typu **Produkt_dto**.

Kod, który należy dodać do interfejsu **Fasada_warstwy_biznesowej_ejbRemote**

@Remote

```
public interface Fasada_warstwy_biznesowej_ejbRemote {  
    public void utworz_produkt(Produkt_dto produkt_dto);  
    public Produkt_dto dane_produktu();  
    public ArrayList<ArrayList<String>> items();  
    public ArrayList<Produkt_dto> items_();  
    public int count();  
    public ArrayList<Produkt_dto> findRange(int[] range);  
    public boolean isStan() ;  
    public void setStan(boolean stan);  
}
```

Kod, który należy dodać do klasy **Fasada_warstwy_biznesowej_ejb**, pełniącej rolę tzw sesyjnej fasady logiki biznesowej, realizowanej przez obiekt typu **Fasada_warstwy_biznesowej**

```
public ArrayList<Produkt_dto> findRange(int[] range) {  
    return fasada.findRange(range);  
}
```

```
public int count() {  
    return fasada.count();  
}
```

```
public void setStan(boolean stan) {  
    fasada.setStan(stan);  
}
```

```
public boolean isStan() {  
    return fasada.isStan();  
}
```

Pośredniczenie w przekazaniu podzbioru danych z logiki biznesowej do aplikacji klienckich: desktopowej i internetowej

Prezentacja stronicowania (1) – należy kolejno wstawić dane nowych produktów za pomocą aplikacji klienckiej desktopowej i internetowej i wyświetlać wynik za pomocą formularzu **lista_produktyw.xhtml** w klienckiej aplikacji internetowej.

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/li Search

Dodaj produkt
Lista produktow

1 .. 1 / 1

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120 zł	10 %	wtorek, 28-03-2017	108 zł

Powrot

Bottom

Dodaj produkt
Lista produktow

1 .. 2 / 2

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120 zł	10 %	wtorek, 28-03-2017	108 zł
2	Produkt2	220 zł	20 %	wtorek, 28-03-2017	176 zł

Powrot

Bottom

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/li/Lista_p Search

Dodaj produkt
Lista produktow

1 .. 3 / 3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120 zł	10 %	wtorek, 28-03-2017	108 zł
2	Produkt2	220 zł	20 %	wtorek, 28-03-2017	176 zł
3	Produkt3	320 zł	25 %	wtorek, 28-03-2017	240 zł

Powrot

Prezentacja stronicowania (2)

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/list

Dodaj produkt
Lista produktow

1 ..3 /4 **Nastepny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120 zł	10 %	wtorek, 28-03-2017	108 zł
2	Produkt2	220 zł	20 %	wtorek, 28-03-2017	176 zł
3	Produkt3	320 zł	25 %	wtorek, 28-03-2017	240 zł

Powrot

Bottom

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lis

Dodaj produkt
Lista produktow

4 ..5 /5 **Popzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
4	Produkt4	440 zł	50 %	wtorek, 28-03-2017	220 zł
5	Produkt5	550 zł	10 %	wtorek, 28-03-2017	495 zł

Powrot

Bottom

Prezentacja stronicowania (3)

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lis

Dodaj produkt
Lista produktow

4 ..6 /6 **Popzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
4	Produkt4	440 zł	50 %	wtorek, 28-03-2017	220 zł
5	Produkt5	550 zł	10 %	wtorek, 28-03-2017	495 zł
6	Produkt6	660 zł	10 %	wtorek, 28-03-2017	594 zł

Powrot

Bottom

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lis

Dodaj produkt
Lista produktow

7 ..7 /7 **Popzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
7	Produkt7	770 zł	10 %	wtorek, 28-03-2017	693 zł

Powrot

Bottom

21

Prezentacja stronicowania (4)

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lis

Dodaj produkt
Lista produktow

4 ..6 /7 **Popzedni 3** **Nastepny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
4	Produkt4	440 zł	50 %	wtorek, 28-03-2017	220 zł
5	Produkt5	550 zł	10 %	wtorek, 28-03-2017	495 zł
6	Produkt6	660 zł	10 %	wtorek, 28-03-2017	594 zł

Powrot

Bottom

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lis

Dodaj produkt
Lista produktow

1 ..3 /7 **Nastepny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120 zł	10 %	wtorek, 28-03-2017	108 zł
2	Produkt2	220 zł	20 %	wtorek, 28-03-2017	176 zł
3	Produkt3	320 zł	25 %	wtorek, 28-03-2017	240 zł

Powrot

Bottom

Widok listy wprowadzonych produktów w aplikacji desktopowej

MenuDemo

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt1	120.0	10	Tue Mar 28 02:00:00 CE...	108.0
2	Produkt2	220.0	20	Tue Mar 28 02:00:00 CE...	176.0
3	Produkt3	320.0	25	Tue Mar 28 02:00:00 CE...	240.0
4	Produkt4	440.0	50	Tue Mar 28 02:00:00 CE...	220.0
5	Produkt5	550.0	10	Tue Mar 28 02:00:00 CE...	495.0
6	Produkt6	660.0	10	Tue Mar 28 02:00:00 CE...	594.0
7	Produkt7	770.0	10	Tue Mar 28 02:00:00 CE...	693.0

Produkty

[1, Produkt1, 120.0, 10, Tue Mar 28 02:00:00 CEST 2017, 108.0]

[1, Produkt1, 120.0, 10, Tue Mar 28 02:00:00 CEST 2017, 108.0]

[2, Produkt2, 220.0, 20, Tue Mar 28 02:00:00 CEST 2017, 176.0]

[3, Produkt3, 320.0, 25, Tue Mar 28 02:00:00 CEST 2017, 240.0]

[4, Produkt4, 440.0, 50, Tue Mar 28 02:00:00 CEST 2017, 220.0]

[5, Produkt5, 550.0, 10, Tue Mar 28 02:00:00 CEST 2017, 495.0]

[6, Produkt6, 660.0, 10, Tue Mar 28 02:00:00 CEST 2017, 594.0]

[7, Produkt7, 770.0, 10, Tue Mar 28 02:00:00 CEST 2017, 693.0]

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
2	Produkt2	220.0	20	Tue Mar 28 02:00:00 CE...	176.0
3	Produkt3	320.0	25	Tue Mar 28 02:00:00 CE...	240.0
4	Produkt4	440.0	50	Tue Mar 28 02:00:00 CE...	220.0
5	Produkt5	550.0	10	Tue Mar 28 02:00:00 CE...	495.0
6	Produkt6	660.0	10	Tue Mar 28 02:00:00 CE...	594.0
7	Produkt7	770.0	10	Tue Mar 28 02:00:00 CE...	693.0

Produkty

[1, Produkt1, 120.0, 10, Tue Mar 28 02:00:00 CEST 2017, 108.0]

[1, Produkt1, 120.0, 10, Tue Mar 28 02:00:00 CEST 2017, 108.0]

[2, Produkt2, 220.0, 20, Tue Mar 28 02:00:00 CEST 2017, 176.0]

[3, Produkt3, 320.0, 25, Tue Mar 28 02:00:00 CEST 2017, 240.0]

[4, Produkt4, 440.0, 50, Tue Mar 28 02:00:00 CEST 2017, 220.0]

[5, Produkt5, 550.0, 10, Tue Mar 28 02:00:00 CEST 2017, 495.0]

[6, Produkt6, 660.0, 10, Tue Mar 28 02:00:00 CEST 2017, 594.0]

[7, Produkt7, 770.0, 10, Tue Mar 28 02:00:00 CEST 2017, 693.0]

1.2. Wprowadzenie warunkowego renderowania strony **rezultat2.xhtml** w celu zapewnienia wieloużywalności tej strony – wyświetlanie danych wybranego produktu

Dodanie do komponentu **dataTable** na stronie **lista_produkow.xhtml** kolumny z przyciskiem obsługującym operację **Rezultat**

Podany znacznik **h:column** należy umieścić jako ostatni w komponencie **dataTable**

```
<h:column>
```

```
<f:facet name="header">
```

```
  <h:outputText value="&nbsp;"/>
```

```
</f:facet>
```

```
<h:commandLink action="#{managed_produkot.prepareView}"  
                value="#{bundle['lista_produkow.rezultat']}/>
```

```
</h:column>
```

Wprowadzenie renderowanych przycisków na stronie **rezultat2.xhtml**
– w przypadku wywołania tej strony ze strony **dodaj_produkt2.xhtml**
zmienna **powrot** jest **równa 1** i następuje powrót do strony
index1.xhtml, a w przypadku wywołania ze strony
lista_produktow.xhtml zmienna **powrot** jest **równa 0** – wtedy ze
strony powraca się z powrotem do strony **lista_produkt.xhtml**

```
<h:commandButton id="powrot1"  
    value="#{bundle['rezultat2.akcja']}"  
    action="/faces/index1"  
    rendered="#{managed_produkt.powrot!=0}"/>  
<h:commandButton id="powrot2"  
    value="#{bundle['rezultat2.akcja']}"  
    action="#{managed_produkt.powrot}"  
    rendered="#{managed_produkt.powrot==0}"/>
```

Dodanie do klasy **Managed_produk**t obsługi przycisku umożliwiającego przejście do strony **rezultat2.xhtml** ze strony **lista_produk**tow.xhtml (metoda **prepareView**) i powrót ze strony **rezultat2.xhtml** do strony **lista_produk**tow.xhtml (metoda **powrot**) oraz atrybut **powrot**

```
private int powrot = 1;
public int getPowrot() {
    return powrot;
}

public String prepareView() {
    produkt_dto = (Produkt_dto) items.getRowData();
    powrot = 0;
    stan=1;
    return "rezultat2";
}

public String powrot() {
    powrot = 1;
    produkt_dto = new Produkt_dto();
    return "lista_produktow";
}
```

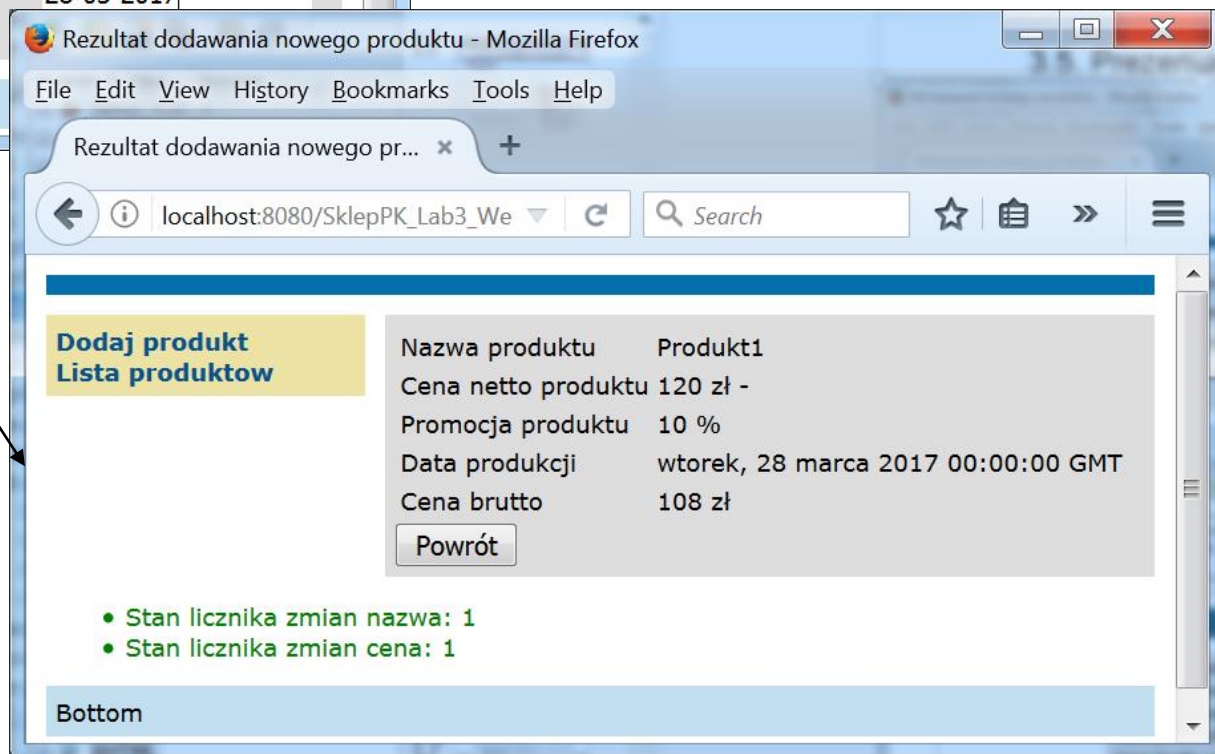
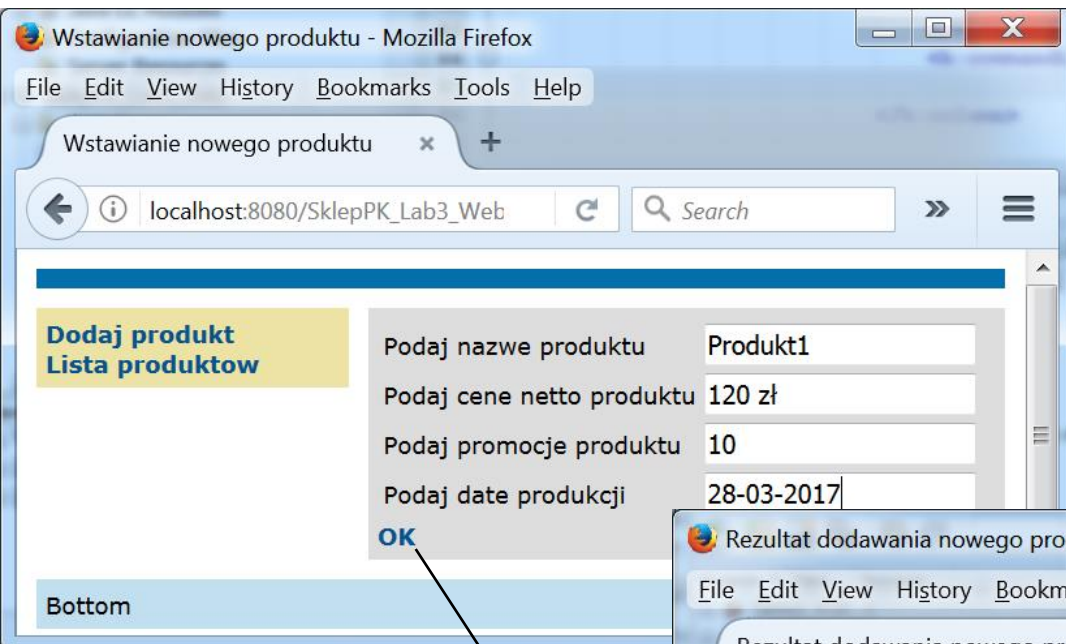
Atrybut **powrot** pozwala renderować przyciski umożliwiające powrót ze strony **rezultat2.xhtml** do strony index1.xhtml (**powrot = 1**) lub **lista_produk**tow.xhtml (**powrot = 0**).

```
public void dodaj_produkt() {  
    fasada.utworz_produkt(produkt_dto);  
    powrot = 1;  
    dane_produktu();  
    recreateModel();  
    getPagination().nextPage();  
}
```

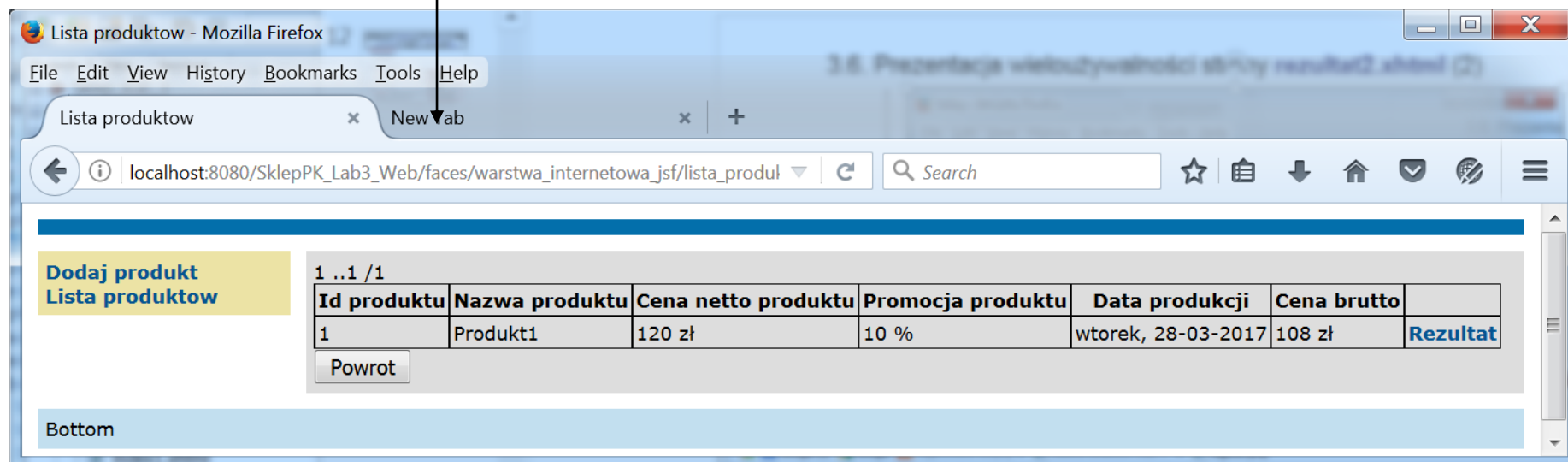
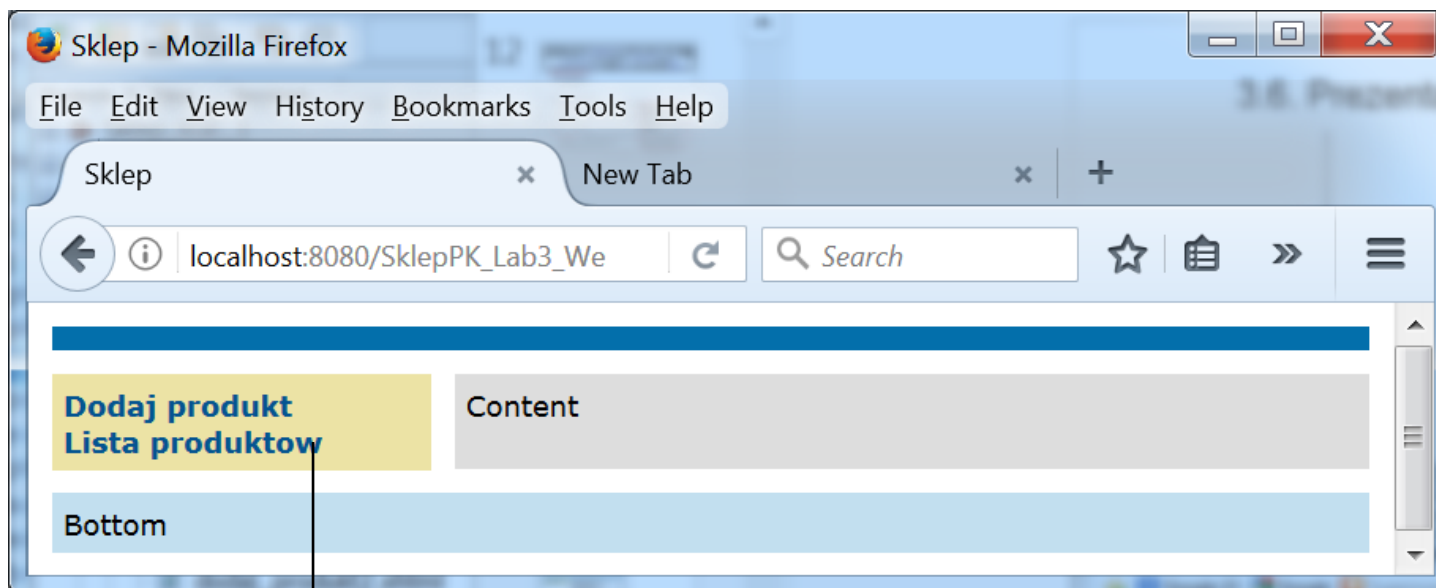
```
public DataModel getItems() {  
    if (items == null || fasada.isStan() ) {  
        items = getPagination().createPageDataModel();  
    }  
    powrot = 1;  
    return items;  
}
```

Atrybut **powrot** ma nadaną wartość 1 w przypadku opuszczenia strony **rezultat2.xhtml** za pomocą bloku „left” szablonu np. i przejście do strony **dodaj_produkt2.xhtml** lub **lista_produktow.xhtml** z renderowaniem właściwych przycisków.

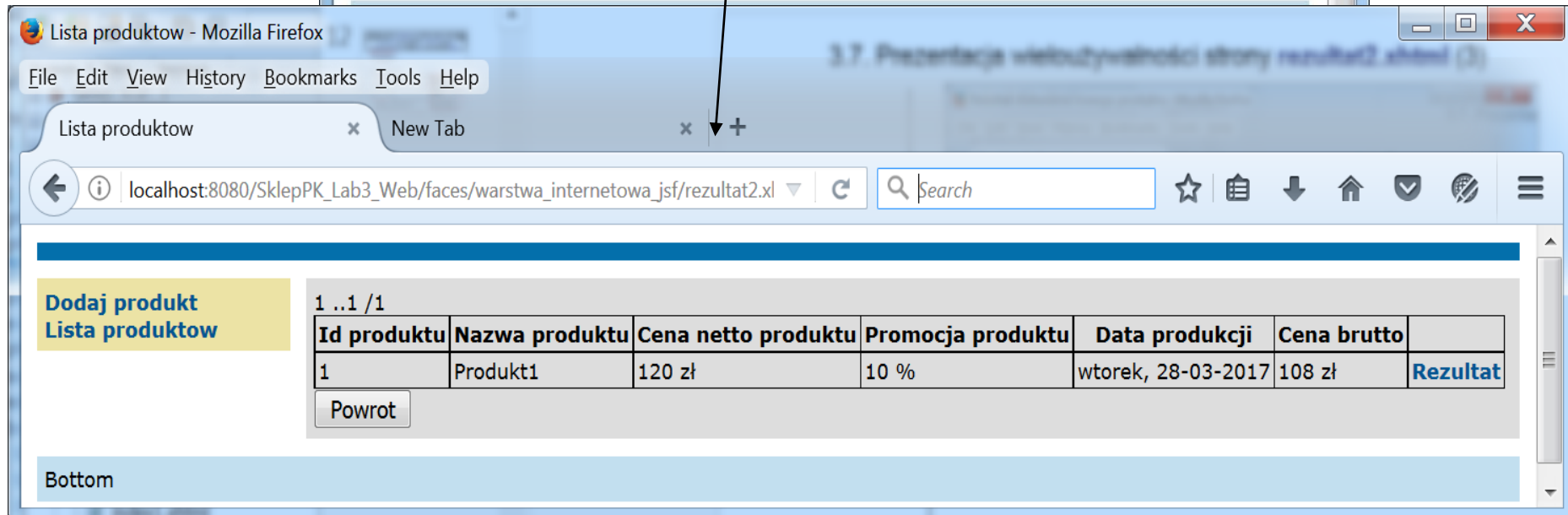
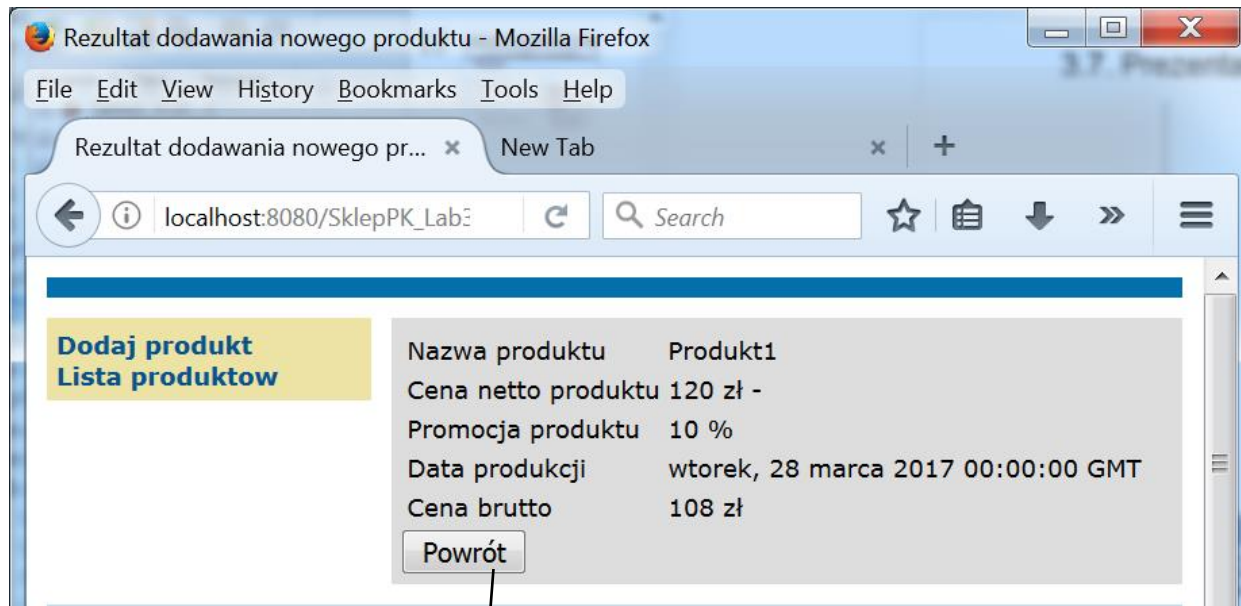
Prezentacja wieloużywalności strony [rezulta2.xhtml](#) (1)- przejście na stronę [rezultat2.xhtml](#) po zakończeniu wprowadzania danych **OK** (3)



2.6. cd. Prezentacja wieloużywalności strony **rezultat2.xhtml** (2) – po kliknięciu na przycisk **Rezultat** na stronie **lista_produkow.xhtml** wywołana jest strona **rezultat2.xhtml** (4)



2.6. cd Prezentacja wieloużywalności strony [rezultat2.xhtml](#) (5) – tak wywołana strona [rezultat2.xhtml](#) ze strony [lista_produkow.xhtml](#) po kliknięciu na przycisk **Powrot** spowoduje powrót na stronę [lista_produkow.xhtml](#)



1.3. Wprowadzenie warunkowego renderowania strony
dodaj_produk2.xhtml w celu
zapewnienia wieloużywalności tej strony
– modyfikacja danych wybranego
produktu

Dodanie przycisku wywołującego stronę **dodaj_produkt2.xhtml** w celu modyfikacji danych produktu z wybranej wiersza tabeli typu **dataTable** na stronie **lista_produktow.xhtml**.

<h:column>

<f:facet name="header">

<h:outputText value=" "/>

</f:facet>

**<h:commandLink action="#{managed_produkt.prepareView}"
value="#{bundle['lista_produktow.rezultat']}" />**

<h:outputText value=" " />

**<h:commandLink action="#{managed_produkt.prepareEdit}"
value="#{bundle['lista_produktow.edycja']}" />**

<h:outputText value=" " />

</h:column>

Dodanie klasy **JsfUtil** do pakietu **pomoc** w projekcie **SklepPK_Lab3_Web**. Dodane metod do klasy **JsfUtil** do obsługi błędów np. metody **update** klasy **Managed_produk**t

```
public static void addErrorMessage(Exception ex, String defaultMsg) {  
    String msg = ex.getLocalizedMessage();  
    if (msg != null && msg.length() > 0) {  
        addErrorMessage(msg);  
    } else {  
        addErrorMessage(defaultMsg);    }  
}
```

```
public static void addErrorMessage(String msg) {  
    FacesMessage facesMsg =  
        new FacesMessage(FacesMessage.SEVERITY_ERROR, msg, msg);  
    FacesContext.getCurrentInstance().addMessage(null, facesMsg);  
}
```

```
public static void addSuccessMessage(String msg) {  
    FacesMessage facesMsg =  
        new FacesMessage(FacesMessage.SEVERITY_INFO, msg, msg);  
    FacesContext.getCurrentInstance().addMessage("successInfo", facesMsg);  
}
```

Kolejny konstruktor, dodany do klasy **Produkt_dto** w celu wyszukania obiektu typu Produkt1 do edycji – wywołana z metody **edit** z klasy **Fasada_warstwy_biznesowej**

```
public Produkt_dto(Produkt_dto o) {  
    nazwa = o.getNazwa();  
    cena = o.getCena();  
    promocja = o.getPromocja();  
    data_produkcji = o.getData_produkcji();  
}
```

Dodanie atrybutu **zmiana** do klasy **Managed_produk**t w celu renderowania strony **dodaj_produk2.xhtml**

@Named(value = "managed_produk")

@SessionScoped

public class Managed_produk implements ActionListener, Serializable {

@EJB

private Fasada_warstwy_biznesowej fasada;

private DataModel items;

private int stan = 1;

private Produkt_dto produkt_dto = new Produkt_dto();

private NumberConverter number_convert = new NumberConverter();

private PaginationHelper pagination;

private int powrot = 1;

private int zmiana = 1;

public int getZmiana() { return zmiana; }

Metoda **prepareEdit**, wywołana po kliknięciu na link **Edycja** na stronie **lista_produkow.xhtml** nadaje wartość 0 zmiennej **zmiana**, co powoduje przejście ze strony **lista_produkow.xhtml** do strony **dodaj_produk2.xhtml** i powrót po edycji do strony **lista_produkow.xhtml**

```
Produkt_dto produkt_dto_przed;
```

```
public String prepareEdit() {  
    produkt_dto = (Produkt_dto) items.getRowData();  
    produkt_dto_przed= new Produkt_dto(produkt_dto);  
    zmiana = 0;  
    return "dodaj_produk2";  
}
```

Obiekt **produkt_dto_przed** zawiera dane przed modyfikacją,

Metoda **update** wywołana ze strony **dodaj_produk2.xhtml**, gdy atrybut zmiana jest równy 0. Jeśli wynik modyfikacji danych jest równy true, zostanie wyświetlony komunikat o dokonanej zmianie danych, a przeciwnym wypadku informacja o błędzie.

```
public String update() {  
    try {  
        boolean wynik=getFasada().edit(produkt_dto_przed, produkt_dto);  
        produkt_dto = new Produkt_dto();  
        zmiana = 1;  
        recreateModel();  
        if(wynik)  
            JsfUtil.addSuccessMessage(ResourceBundle.getBundle("/Bundle").  
                getString("Produkt_zmieniony"));  
        else throw new Exception();  
        return "lista_produkow";  
    } catch (Exception e) {  
        JsfUtil.addErrorMessage(e,  
            ResourceBundle.getBundle("/Bundle").getString("Blad_modyfikacji"));  
        return "lista_produkow"; }  
    }  
}
```

```
public DataModel getItems() {  
    if (items == null || fasada.isStan()) {  
        items = getPagination().createPageDataModel();  
    }  
    zmiana = 1;  
    powrot = 1;  
    return items;  
}
```

```
public void dodaj_produkt() {  
    fasada.utworz_produkt(produkt_dto);  
    powrot = 1;  
    zmiana=1;  
    dane_produktu();  
    recreateModel();  
    getPagination().nextPage();  
}
```

Zmiany na stronie `dodaj_produkt_2.xhtml`

```
<h:inputText
  id="nazwa"
  title="#{bundle['jsf.dodaj_produkt2.podaj_nazwa']}"
  value="#{managed_produkt.nazwa}"
  required="true,,
  requiredMessage="#{bundle['jsf.dodaj_produkt2.podaj_nazwa_blad']}"
  disabled="#{managed_produkt.zmiana==0}" >
</h:inputText>
```

Dodanie atrybutu **disabled** pozwala pokazać widok komponentu na stronie jako nieaktywny – nie można wprowadzić nazwy produktu

```
<h:commandLink action="rezultat2" value="#{bundle['dodaj_produkt2.akcja']}"
  rendered="#{managed_produkt.zmiana==1}">
  <f:actionListener binding="#{managed_produkt}"/>
</h:commandLink>
<h:commandLink action="#{managed_produkt.update}"
  value="#{bundle['dodaj_produkt2.akcja']}"
  rendered="#{managed_produkt.zmiana==0}"/>
```

Dodawanie nowego produktu

Modyfikacja danych produktu metodą **update**

Modyfikacja metod w klasie **Fasada_warstwy_biznesowej**.

Metoda **edit** w klasie **Fasada_warstwy_biznesowej** wywołana podczas obsługi zdarzenia kliknięcia na przycisk OK. na stronie **dodaj_produkt2.xhtml**, gdy renderowany jest przycisk dla wartości zmiennej **zmiana=0** i wywołana jest metoda **update** z klasy **Managed_produkt**.

Metoda **istnieje_produkt** sprawdza, czy dane przed edycją istnieją w kolekcji **produkty** - jeśli nie istnieją, modyfikacja kończy się błędem. Drugie sprawdzenie za pomocą metody **istnieje_produkt** sprawdza, czy po modyfikacji dane nadal są unikatowe - jeśli nie, modyfikacja kończy się błędem.

```
int istnieje_produkt(Produkt_dto pdto) {  
    Produkt1 pom1 = this.wykonaj_produkt(pdto);  
    return getProdukty().indexOf(pom1);  
}
```

```
public boolean edit(Produkt_dto o_przed, Produkt_dto o_update) {  
    int idx1, idx2;  
    stan = true;  
    idx1=this.istnieje_produkt(o_przed);  
    if(idx1==-1)                //taki produkt do edycji nie istnieje  
        return false;  
    idx2=this.istnieje_produkt(o_update);  
    if(idx2!=-1)                //nie mozna modyfikowac, bo juz taki produkt istnieje  
        return false;  
    Produkt1 p = getProdukty().get(idx1);  
    p.setCena(o_update.getCena());  
    p.setData_produkcji(o_update.getData_produkcji());  
    p.setPromocja(o_update.getPromocja());  
    return true;  
}
```

Uzupełnienie deklaracji metod w interfejsie logiki biznesowej – klasa **SklepPK_interfejs_1**

```
package warstwa_biznesowa_ejb;

import java.util.ArrayList;
import javax.ejb.Remote;
import warstwa_biznesowa.dto.Produkt_dto;
@Remote
public interface Fasada_warstwy_biznesowej_ejbRemote {
    public void utworz_produkt(Produkt_dto produkt_dto);
    public Produkt_dto dane_produktu();
    public ArrayList<ArrayList<String>> items();
    public ArrayList<Produkt_dto> items_();
    public int count();
    public ArrayList<Produkt_dto> findRange(int[] range);
    public boolean isStan();
    public void setStan(boolean stan);
    public boolean edit(Produkt_dto o_przed, Produkt_dto o_update);
}
```

cd. Implementacja nowej metody **edit** w komponencie EJB –
Fasada_warstwy_biznesowej_ejb

```
public boolean edit(Produkt_dto o_przed, Produkt_dto o_update) {  
    return fasada.edit(o_przed, o_update);  
}
```

Prezentacja procesu modyfikacji danych (1) – próba naruszenia interalności danych podczas modyfikacji danych

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtm

Dodaj produkt
Lista produktow

1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	Rezultat Edycja Usun
3	Produkt4	220 zł	20 %	piątek, 28-04-2017	176 zł	Rezultat Edycja Usun

Powrot

Bottom

Wstawianie nowego produktu - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wstawianie nowego produktu

localhost:8080/SklepPK_Lab3_We

Dodaj produkt
Lista produktow

Podaj nazwe produktu

Podaj cene netto produktu

Podaj promocje produktu

Podaj date produkcji

OK

Bottom

Wstawianie nowego produktu - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wstawianie nowego produktu

localhost:8080/SklepPK_Lab:

Dodaj produkt
Lista produktow

Podaj nazwe produktu

Podaj cene netto produktu

Podaj promocje produktu

Podaj date produkcji

OK

Bottom

cd. Prezentacja procesu modyfikacji danych (2) – próba wykonania zmian, które naruszają integralność danych

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/dodaj_produk2.xhtml

Dodaj produkt
Lista produktow

1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	Rezultat Edycja Usun
3	Produkt4	220 zł	20 %	piątek, 28-04-2017	176 zł	Rezultat Edycja Usun

Powrot

- Stan licznika zmian cena: 3
- Nie dokonano zmiany danych produktu

Bottom

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produk2.xhtml

Dodaj produkt
Lista produktow

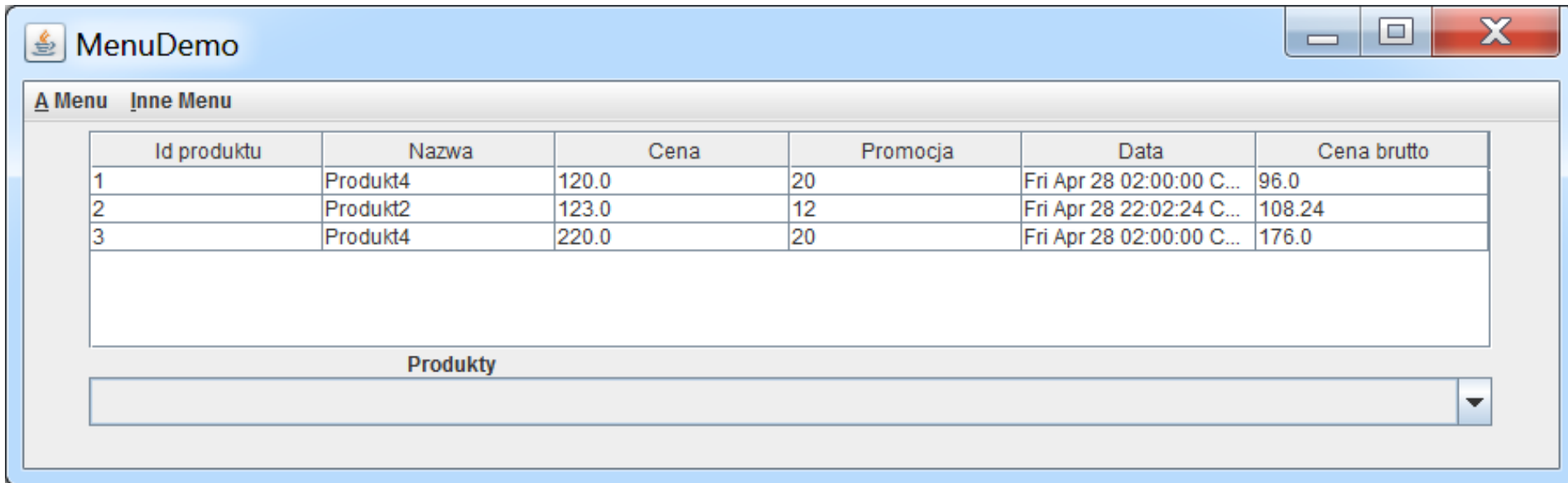
1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	Rezultat Edycja Usun
3	Produkt4	220 zł	20 %	piątek, 28-04-2017	176 zł	Rezultat Edycja Usun

Powrot

Bottom

cd. Prezentacja procesu modyfikacji danych (3)



Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt4	120.0	20	Fri Apr 28 02:00:00 C...	96.0
2	Produkt2	123.0	12	Fri Apr 28 22:02:24 C...	108.24
3	Produkt4	220.0	20	Fri Apr 28 02:00:00 C...	176.0

Produkty

cd. Prezentacja procesu modyfikacji danych (4) – poprawna modyfikacja danych

Wstawianie nowego produktu - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wstawianie nowego produktu x +

localhost:8080/SklepPK_Lab3_We Search

Dodaj produkt
Lista produktów

Podaj nazwe produktu Produkt4

Podaj cene netto produktu 220 zł

Podaj promocje produktu 20

Podaj date produkcji 28-04-2017

OK

Bottom

Wstawianie nowego produktu - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wstawianie nowego produktu x +

localhost:8080/SklepPK_Lab3_Wet Search

Dodaj produkt
Lista produktów

Podaj nazwe produktu Produkt4

Podaj cene netto produktu 440 zł

Podaj promocje produktu 30

Podaj date produkcji 28-04-2017

OK

Bottom

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/dodaj_produk2.xhtml Search

Dodaj produkt
Lista produktów

1 ..3 /3

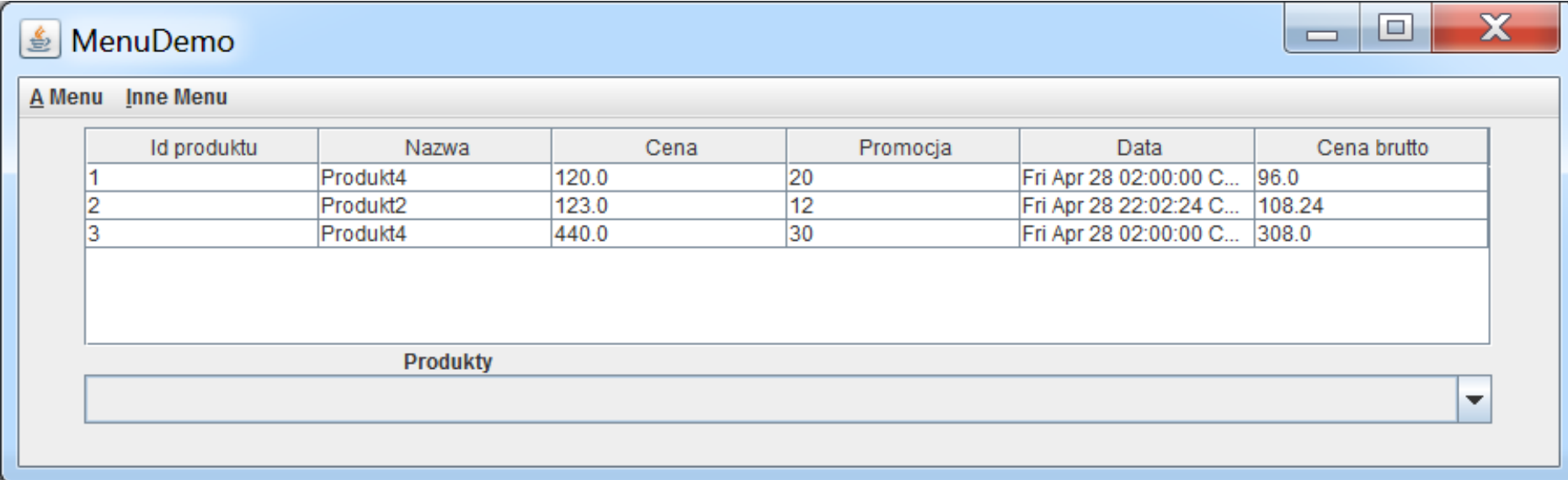
Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	Rezultat Edycja Usun
3	Produkt4	440 zł	30 %	piątek, 28-04-2017	308 zł	Rezultat Edycja Usun

Powrot

- Stan licznika zmian cena: 4
- Zmiana danych produktu

Bottom

cd. Prezentacja procesu modyfikacji danych (5) – poprawna modyfikacja danych



MenuDemo

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt4	120.0	20	Fri Apr 28 02:00:00 C...	96.0
2	Produkt2	123.0	12	Fri Apr 28 22:02:24 C...	108.24
3	Produkt4	440.0	30	Fri Apr 28 02:00:00 C...	308.0

Produkty

1.4. Dodanie na stronie `lista_produkow.xhtml` przycisku do usuwania danych

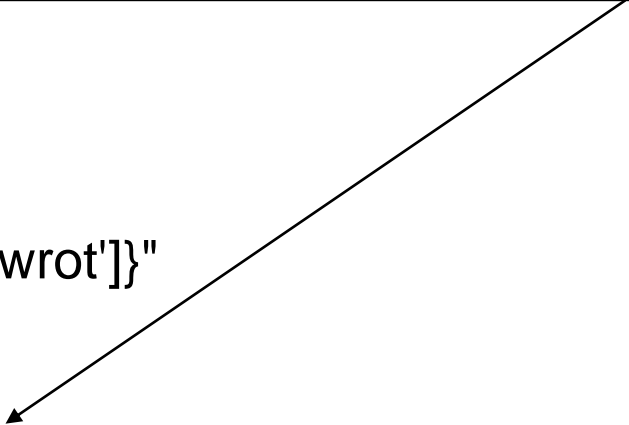
Dodanie przycisku do usuwania produktu wybranego w wierszu tabeli typu **dataTable** na stronie **lista_produkow.xhtml**

```
<h:column>
<f:facet name="header">
    <h:outputText value="&nbsp;" />
</f:facet>
<h:commandLink action="#{managed_produk.prepareView}"
                value="#{bundle['lista_produkow.rezultat']}" />
<h:outputText value=" " />
<h:commandLink action="#{managed_produk.prepareEdit}"
                value="#{bundle['lista_produkow.edycja']}" />
<h:outputText value=" " />
<h:commandLink action="#{managed_produk.destroy}"
                value="#{bundle['lista_produkow.usun']}" />
</h:column>
```

Dodanie przycisku do aktualizacji stronicowania widoku **dataTable** na stronie **lista_produkow.xhtml** w przypadku usuwania wierszy przez inną instancję klienta internetowego

```
<h:commandButton id="powrot"  
    value="#{bundle['lista_produkow.powrot']}"  
    action="/faces/index1" />
```

```
<h:commandButton id="refresh"  
    value="#{bundle['lista_produkow.refresh']}"  
    action="#{managed_produkow.refresh}" />
```



Dodanie metod do obsługi usuwania danych w klasie
Managed_produk

```
public String destroy() {  
    produkt_dto = (Produkt_dto) items.getRowData();  
    int ile = items.getRowCount();  
    if (ile == 1) {  
        this.getPagination().previousPage();  
    }  
    performDestroy();  
    return "lista_produkto";  
}
```

Przygotowanie
aktualizacji strony po
usunięciu wybranego
wiersza tabeli

```
private void performDestroy() {  
    try {  
        getFasada().remove(produkt_dto);  
        JsfUtil.addSuccessMessage(ResourceBundle.getBundle("/Bundle").getString("Usunieto_produk"));  
    } catch (Exception e) {  
        JsfUtil.addErrorMessage(e, ResourceBundle.getBundle("/Bundle").getString("Blad_usuwania"));  
    }  
}
```

Dodanie metody **refresh** w klasie **Managed_produk**t do obsługi przycisku (p.4.2.) do aktualizacji strony **lista_produk**tow.xhtml w przypadku usuwania wierszy przez inną instancję klienta internetowego

```
public String refresh() {  
    getPagination().updatePage();  
    items = getPagination().createPageDataModel();  
    return "lista_produk
```

tow";
}

Aktualizacja modelu komponentu dataTable po zmodyfikowaniu numeru strony, koniecznym w przypadku usunięcia tylu wierszy przez inną instancję klienta internetowego, że zmniejszyła się liczba stron koniecznych do wyświetlenia zawartości tabeli

Dodanie metody **updatePage()** w klasie **PaginationHelper** do aktualizacji numeru ostatniej strony zawierającej ostatni fragment widoku **dataTable** ze strony **lista_produkow.xhtml** w przypadku usuwania wierszy przez inną instancję klienta internetowego

```
public void setPage() {  
    page=getItemsCount() / pageSize;  
}
```

```
public void updatePage() {  
    setPage();  
    int pom2=getItemsCount()%pageSize;  
    if (pom2==0)  
        page--;  
}
```

Aktualizacja numeru ostatniej strony do wyświetlenia ostatniego fragmentu modelu komponentu **dataTable**

Dodanie metody **remove** do klasy **Fasada_warstwy_biznesowej**

```
public void remove(Produkt_dto p) {  
    Produkt1 produkt = wykonaj_produkt(p);  
    getProdukty().remove(produkt);  
}
```


Uzupełnienie deklaracji metod w interfejsie logiki biznesowej – klasa **SklepPK_interfejs_1**

```
package warstwa_biznesowa_ejb;

import java.util.ArrayList;
import javax.ejb.Remote;
import warstwa_biznesowa.dto.Produkt_dto;

@Remote
public interface Fasada_warstwy_biznesowej_ejbRemote {
    public void utworz_produkt(Produkt_dto produkt_dto);
    public Produkt_dto dane_produktu();
    public ArrayList<ArrayList<String>> items();
    public ArrayList<Produkt_dto> items_();
    public int count();
    public ArrayList<Produkt_dto> findRange(int[] range);
    public boolean isStan();
    public void setStan(boolean stan);
    public boolean edit(Produkt_dto o_przed, Produkt_dto o_update);
    public void remove(Produkt_dto p);
}
```

cd. Implementacja nowej metody **remove** w komponencie EJB –
Fasada_warstwy_biznesowej_ejb

```
public void remove(Produkt_dto p) {  
    fasada.remove(p);  
}
```

Prezentacja usuwania produktu (1) – wstawienie 5 produktów, gdzie ostatnie dwa wprowadzono na stronie internetowej (**klient1**)

MenuDemo

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt1	120.0	20	Fri Apr 28 02:00:00 C...	96.0
2	Produkt2	120.0	20	Fri Apr 28 02:00:00 C...	96.0
3	Produkt3	120.0	20	Fri Apr 28 02:00:00 C...	96.0
4	Produkt4	120.0	20	Fri Apr 28 02:00:00 C...	96.0
5	Produkt5	120.0	20	Fri Apr 28 02:00:00 C...	96.0

Produkty

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtm Search

[Dodaj produkt](#)
[Lista produktow](#)

4 ..5 /5 [Poprzedni 3](#)

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
4	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
5	Produkt5	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

[Powrot](#)

[Odśwież stronę](#)

Bottom

cd. Widok danych na stronie **lista_produkow** w drugiej instancji klienta internetowego (**klient2**) – domyślny i po kliknięciu na przycisk **Następny 3** (2)

1 ..3 /5 **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

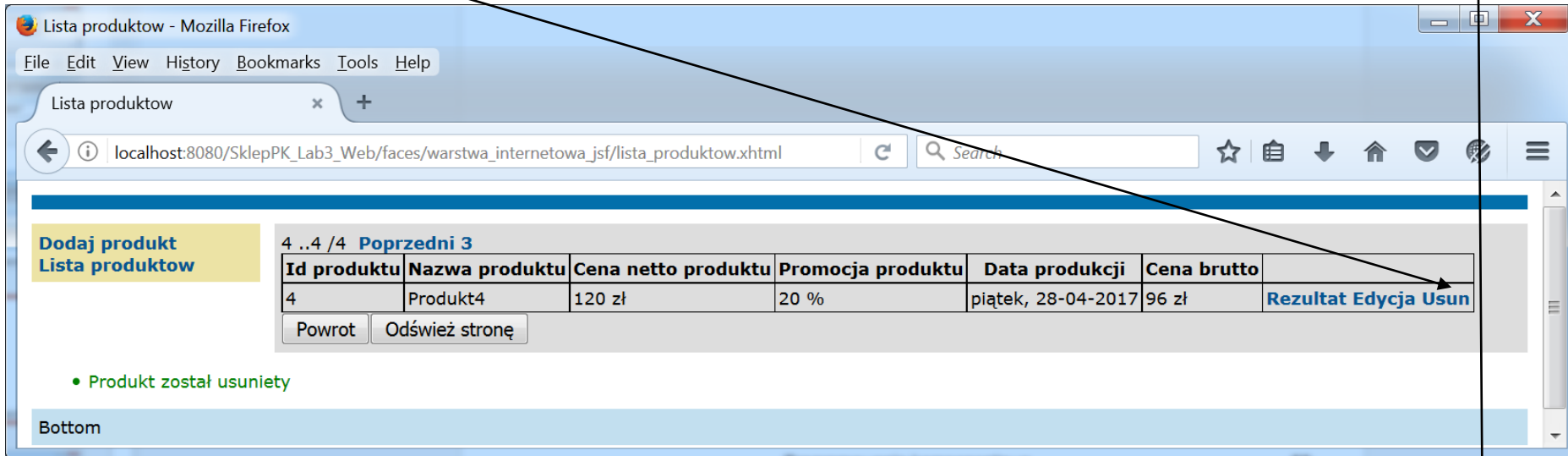
Bottom

4 ..5 /5 **Poprzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
4	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
5	Produkt5	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

Bottom

cd. Usuwanie kolejnych produktów na ostatniej stronie **klienta1** (3). Po usunięciu pozycji 4 automatycznie uaktualniła się strona wyświetlanej tabeli na stronie **lista_produkow**.



Lista produktow - Mozilla Firefox

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

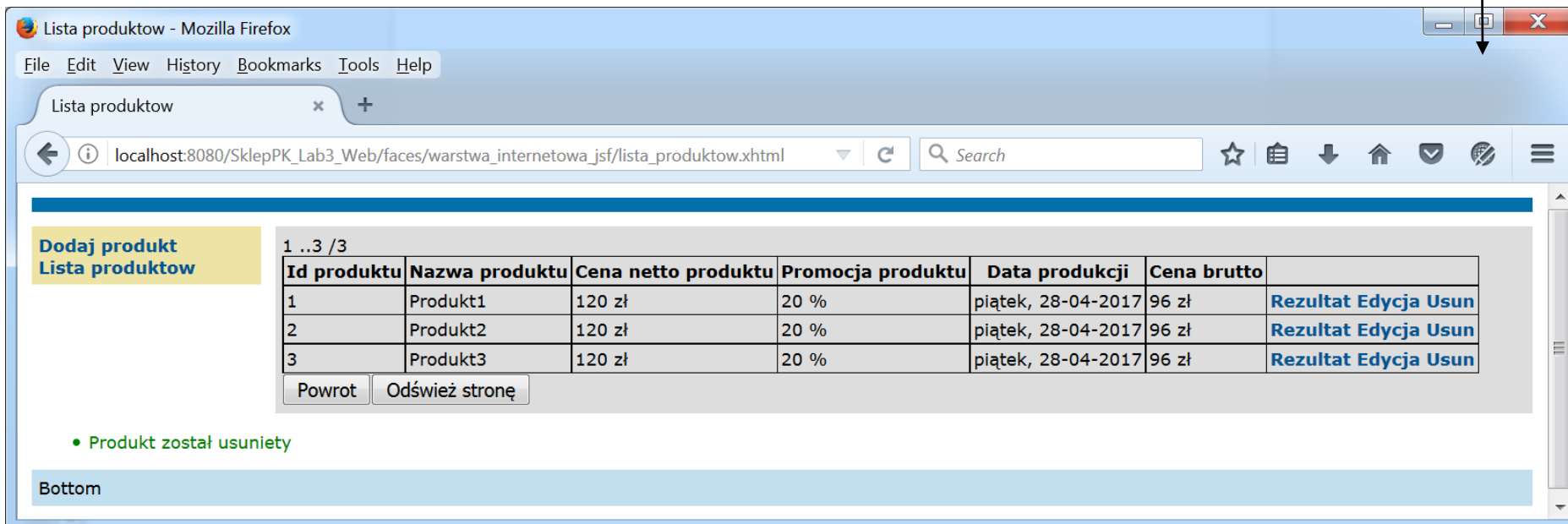
Dodaj produkt
Lista produktow

4 ..4 /4 **Poprzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
4	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

• Produkt został usuniety

Bottom



Lista produktow - Mozilla Firefox

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktow

1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

• Produkt został usuniety

Bottom

cd. Widok nieaktualnej zawartości strony **lista_produkow** w instancji internetowego **klienta2** (4) – i po aktualizacji za pomocą przycisku **Odśwież stronę**

The image displays two screenshots of a web browser window showing a product list. The browser address bar shows the URL: `http://localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa.jsf/li: Lista produktow`.

Top Screenshot (Initial State):

- Page title: Lista produktow
- Navigation: Dodaj produkt, Lista produktow
- Page indicator: 4 ..5 /5 Poprzedni 3
- Table data:

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
4	Produkt4	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
5	Produkt5	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

Buttons: Powrot, Odśwież stronę

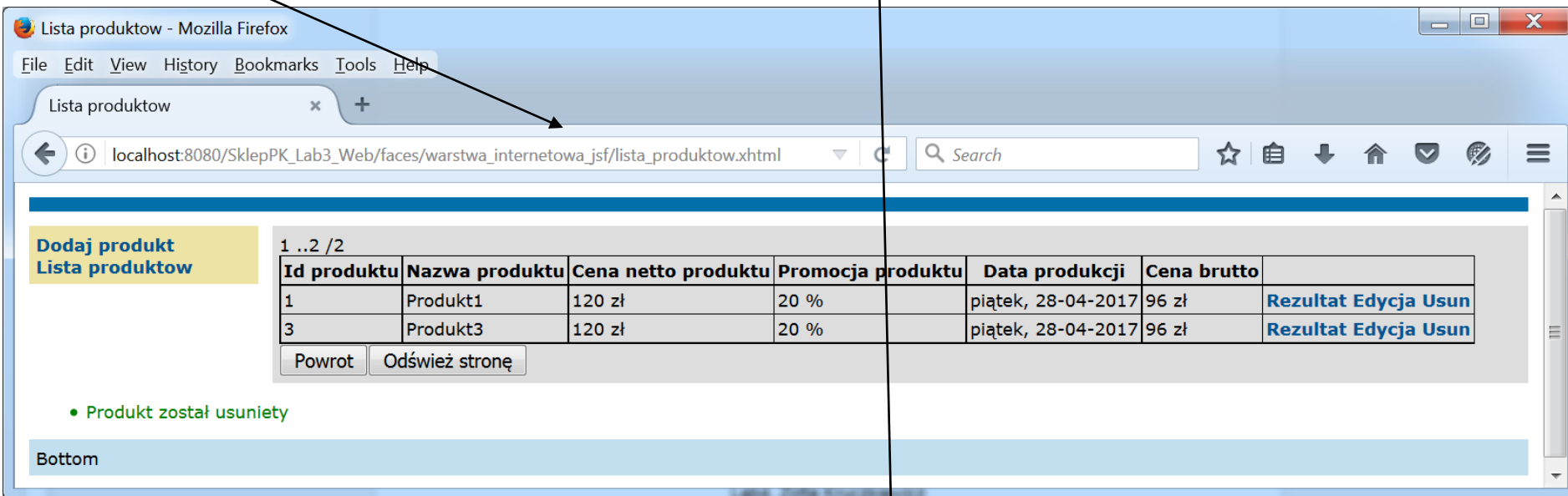
Bottom Screenshot (After Refresh):

- Page title: Lista produktow
- Page indicator: 1 ..3 /3
- Table data:

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

Buttons: Powrot, Odśwież stronę

cd. Kolejne usuwanie wiersza tabeli na stronie **lista_produkow klienta1**. Strona **lista_produkow klienta2** wymaga aktualizacji. (5)



Lista produktow - Mozilla Firefox

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktow

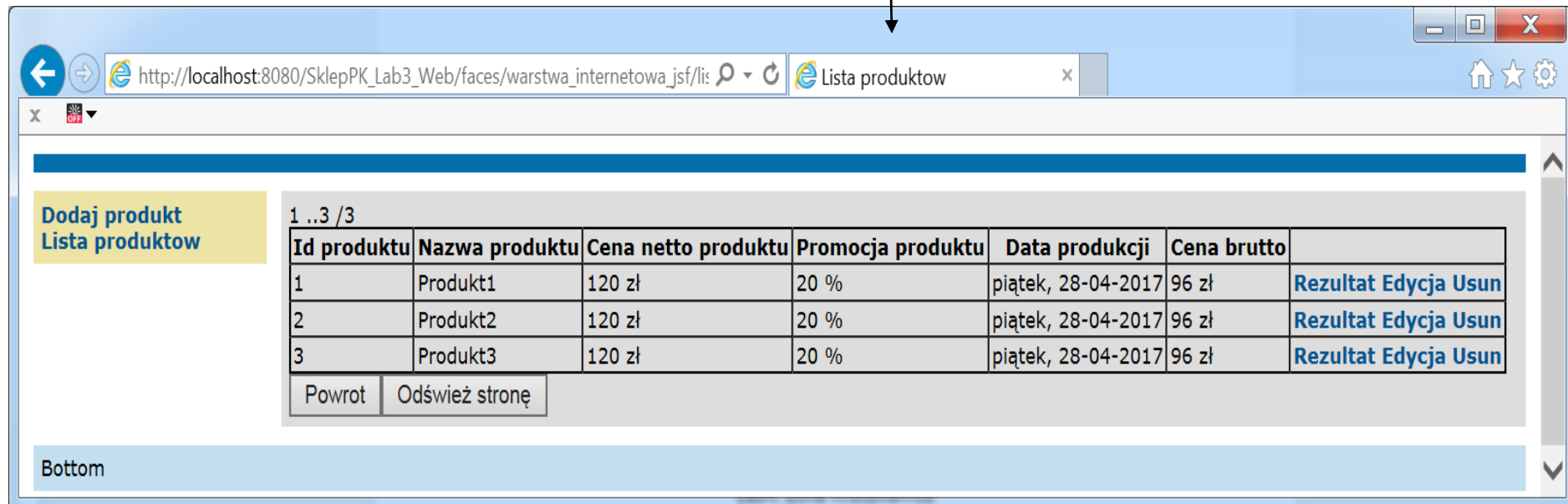
1 .. 2 / 2

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

• Produkt został usuniety

Bottom



http://localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/li: Lista produktow

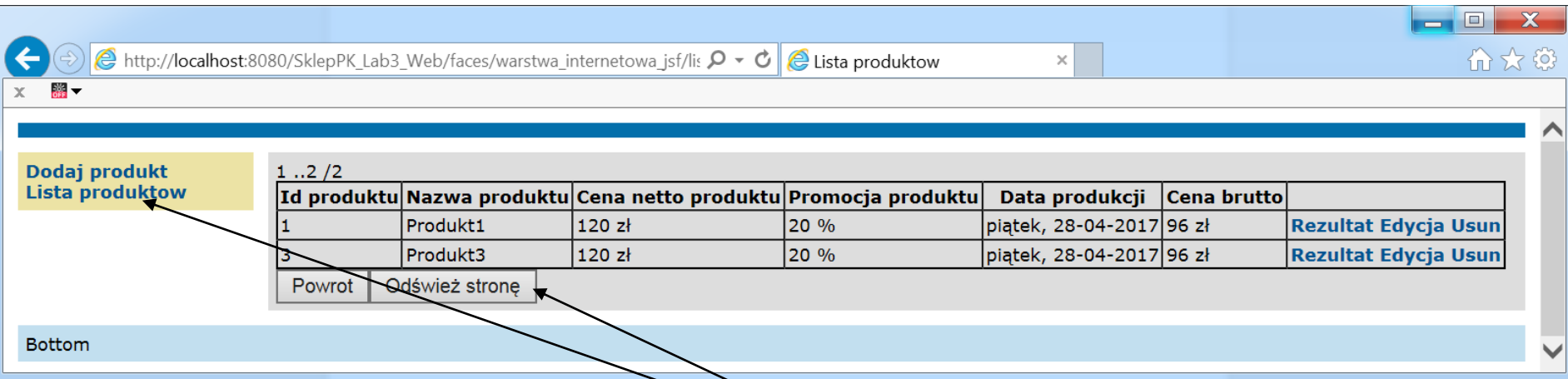
Dodaj produkt
Lista produktow

1 .. 3 / 3

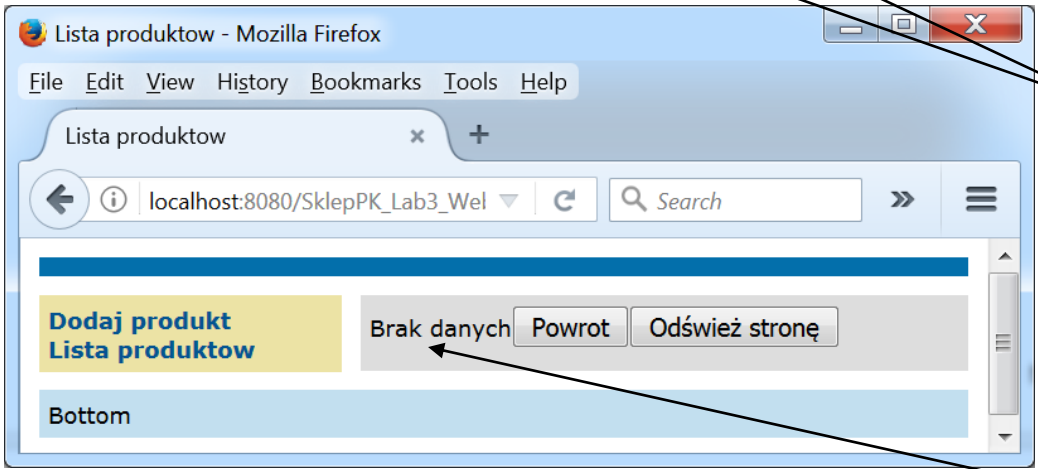
Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

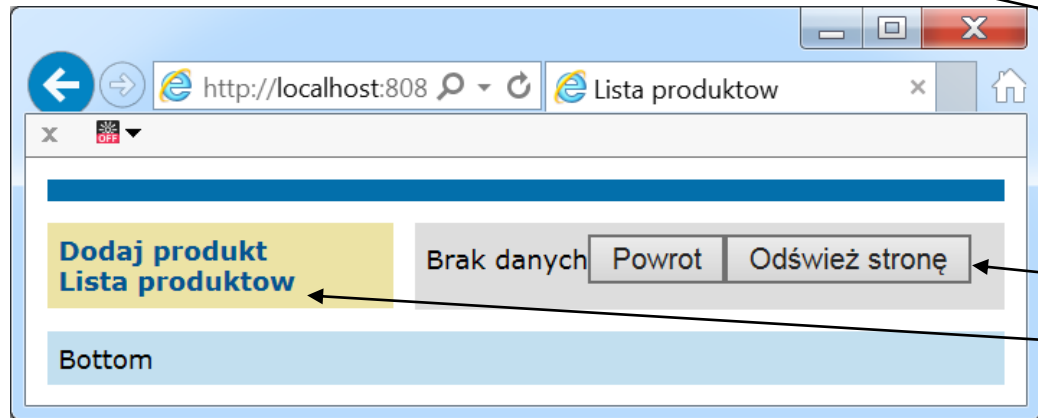
Bottom



cd. Strona **lista_produkow klienta2** może być teraz zaktualizowana za pomocą przycisku **Odśwież stronę** lub **Lista produktow** (6)

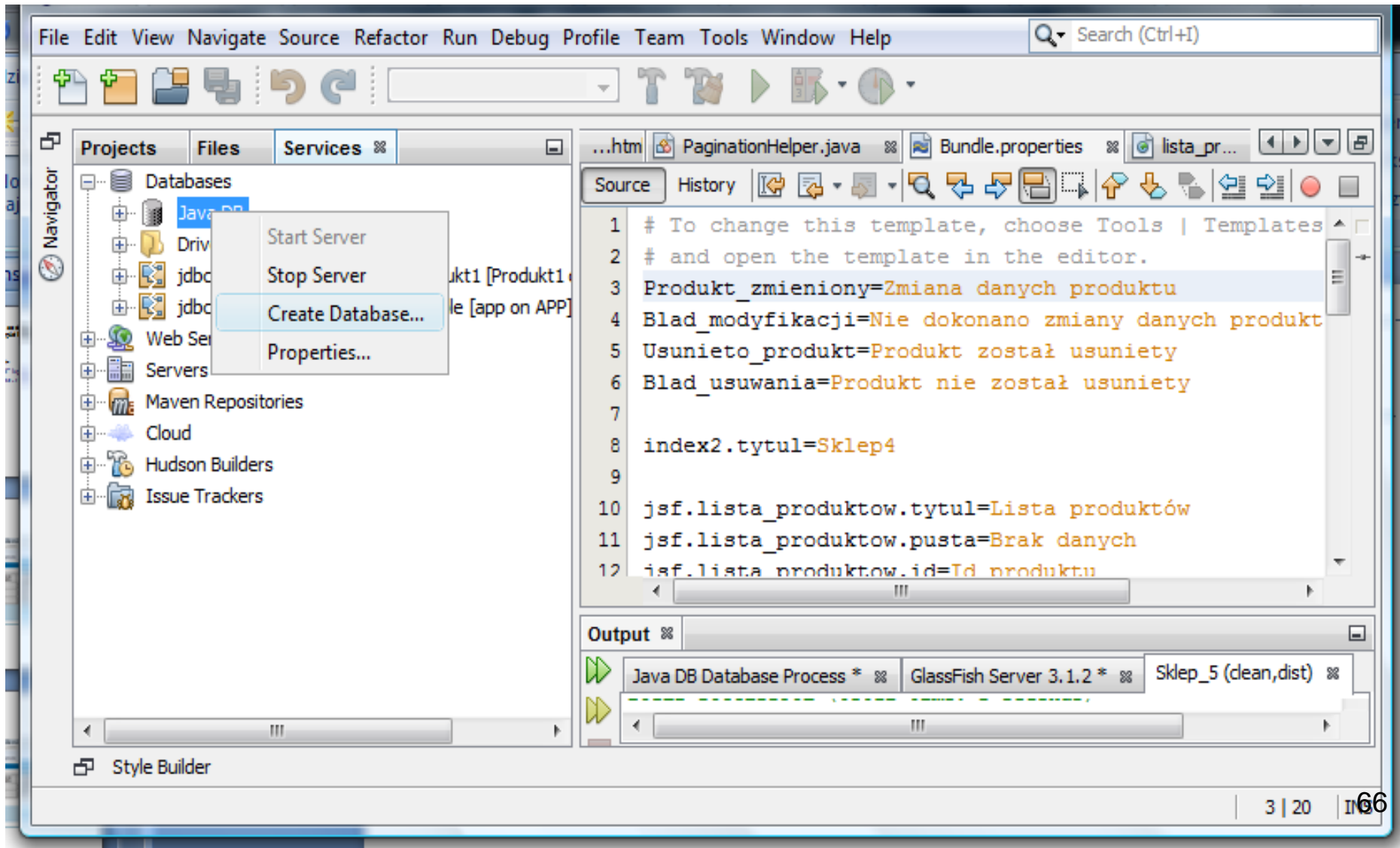


cd. Strona **lista_produkow klienta1** po usunięciu wszystkich danych, automatycznie zaktualizowana. Strona **lista_produkow klienta2** może być teraz zaktualizowana za pomocą przycisku **Odśwież stronę** lub **Lista produktow** (7)



1.5. Utrwalenie danych obiektów typu Produkt1 za pomocą technologii JPA (wykład 4)

Należy utworzyć pustą bazę danych w zakładce **Services** (prawy przycisk myszy na pozycji JavaDB po rozwinięciu pozycji Database i wybór **Create Database**)



Wpisanie nazwy bazy danych (**Database Name**), loginu (**User Name**) i hasła (**Password**)

Create Java DB Database

Database Name: Produkt

User Name: Produkt

Password: ●●●●●●

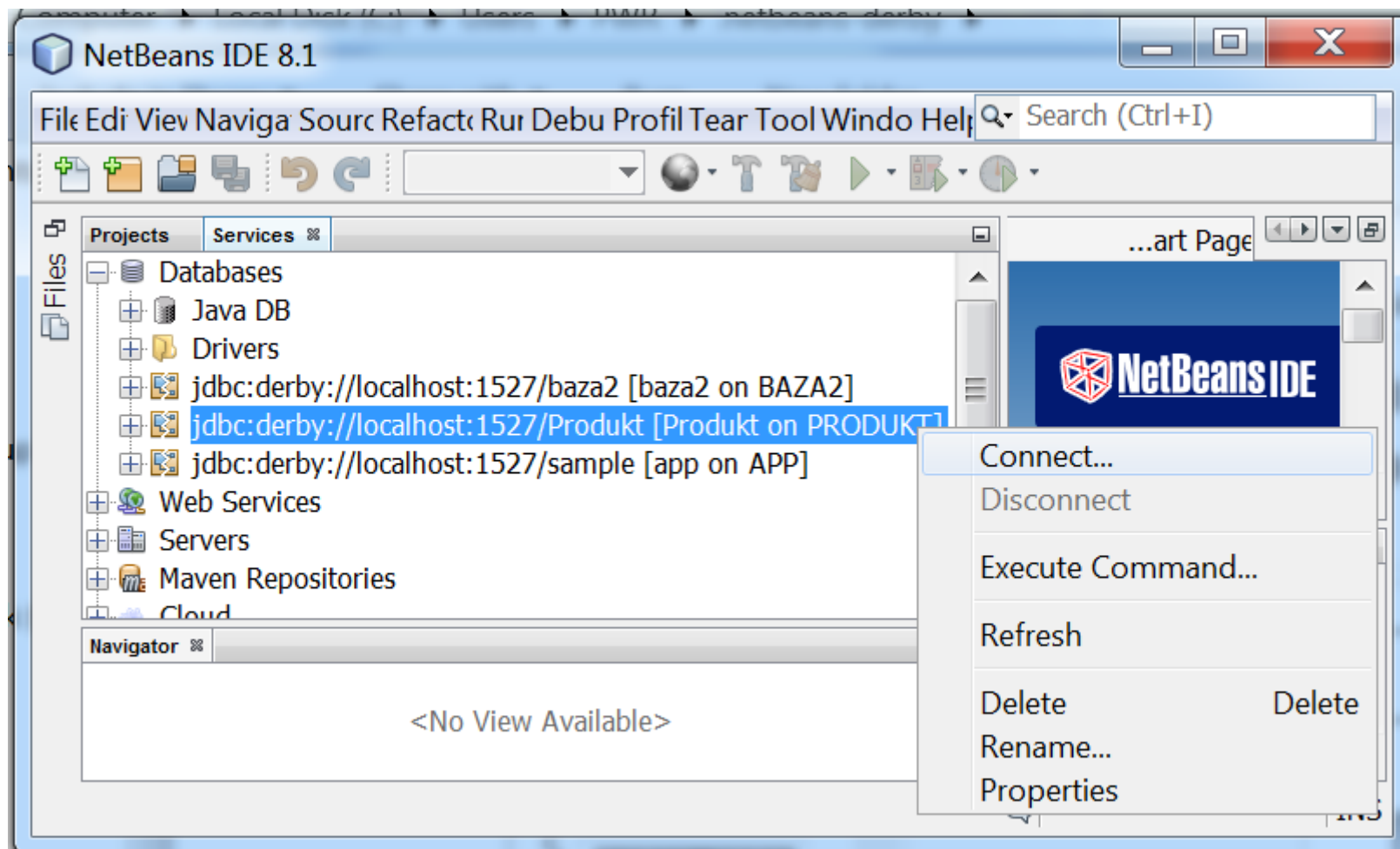
Confirm Password: ●●●●●●

Database Location: C:\Users\kruczkiewicz\.netbeans-derby

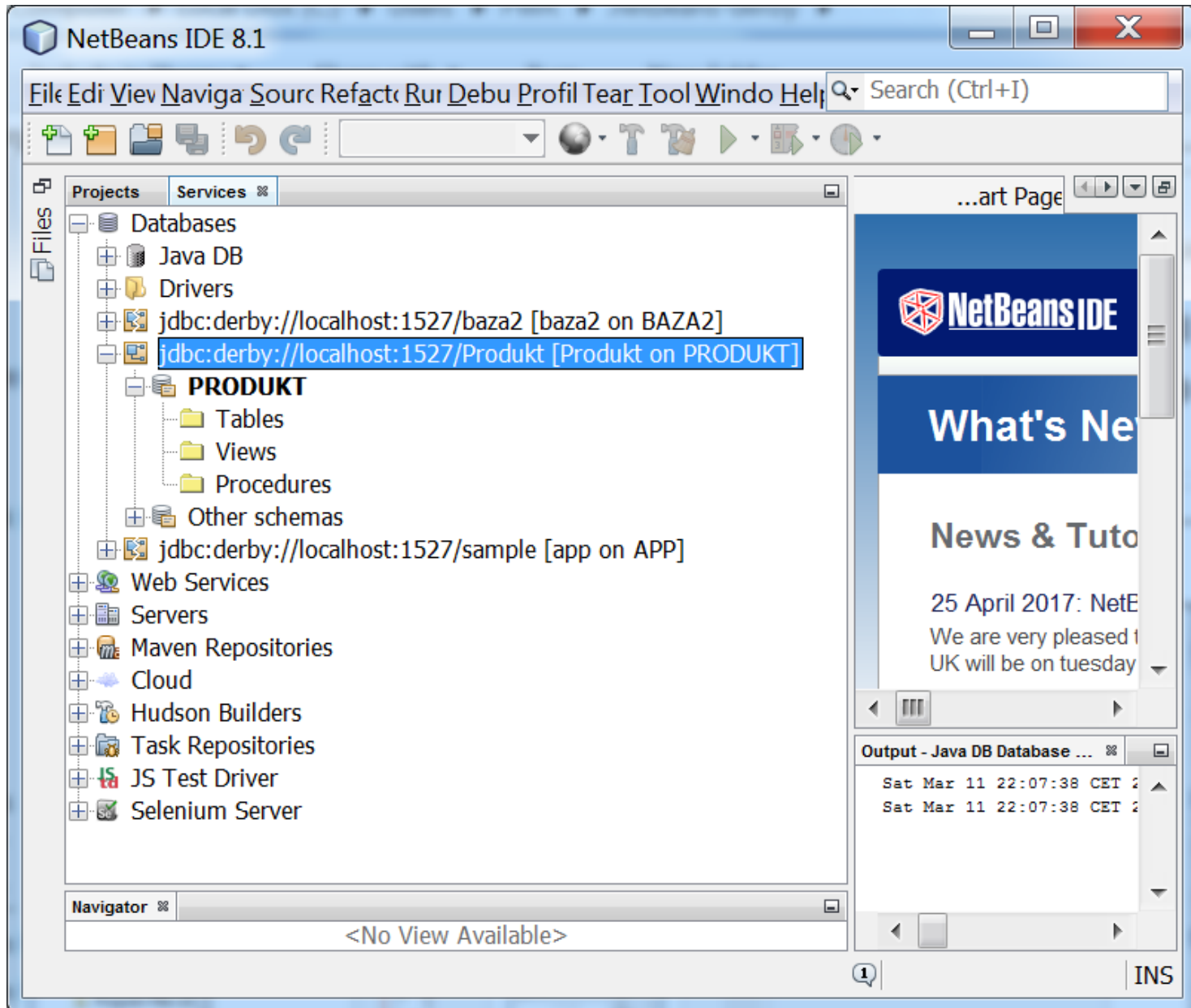
Properties...

OK Cancel

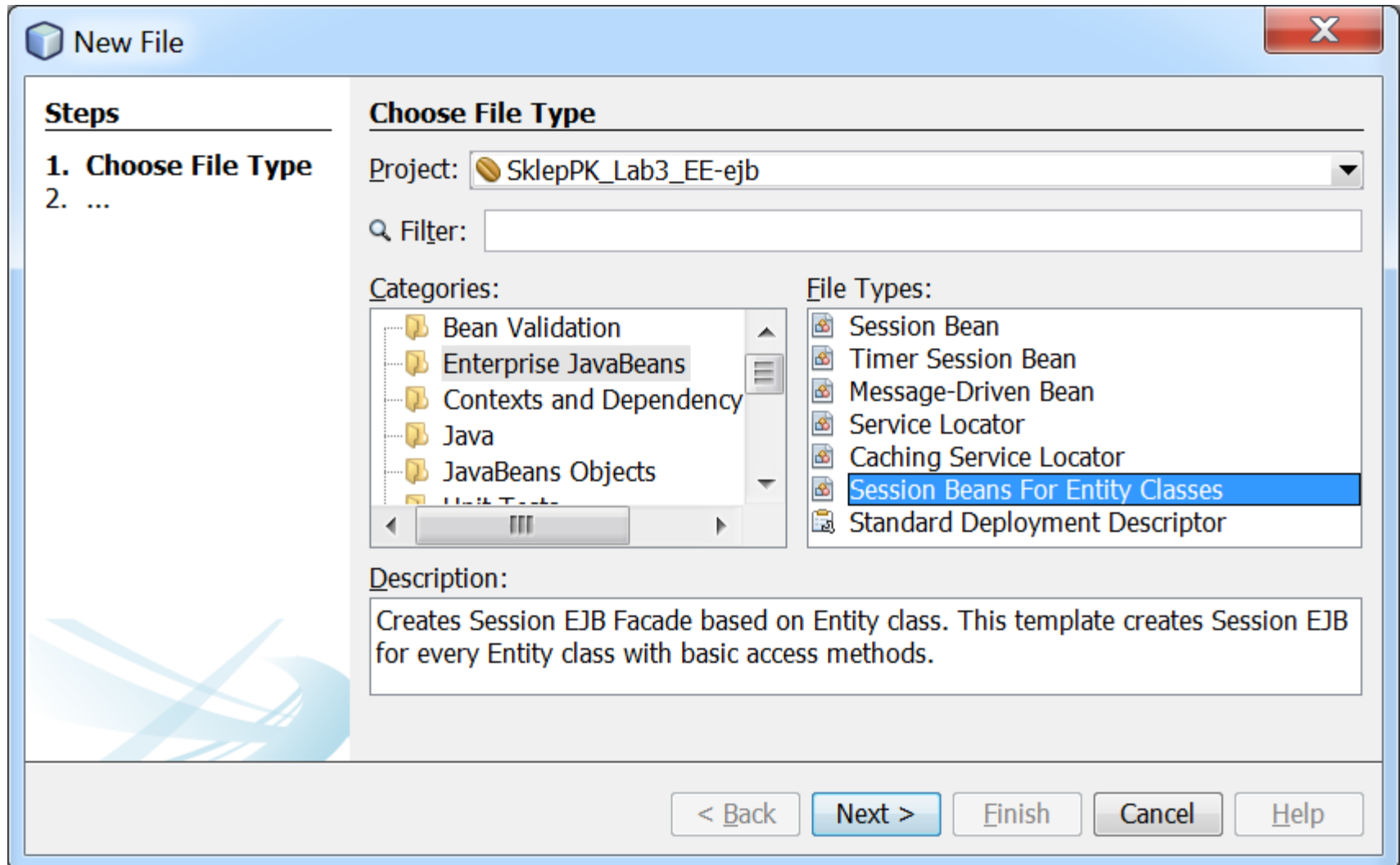
Połączenie z pustą bazą danych (wybór pozycji **Connect** po kliknięciu prawym klawiszem myszy na połączeniu do bazy danych)



Widok utworzonej pustej bazy danych **Produkt**



Dodanie komponentów do utrwalania obiektów typu Produkt1 – New/OtherEnterprise JavaBeans/SessionBeans for Entity Classes



Wybór klasy Produkt1 typu Entity jako komponentu do utrwalania

New Session Beans For Entity Classes

Steps

1. Choose File Type
- 2. Entity Classes**
3. Generated Session Beans
4. Persistence Unit Provider and Database

Entity Classes

Available Entity Classes:

warstwa_biznesowa.entity.Produkt1

Add >

< Remove


Add All >>

<< Remove All

Selected Entity Classes:

Include Referenced Classes

Create Persistence Unit

 Select entity classes.

< Back Next > Finish Cancel Help

New Session Beans For Entity Classes

Steps

1. Choose File Type
- 2. Entity Classes**
3. Generated Session Beans
4. Persistence Unit Provider and Database

Entity Classes

Available Entity Classes:

Add >

< Remove

Add All >>

<< Remove All

Selected Entity Classes:

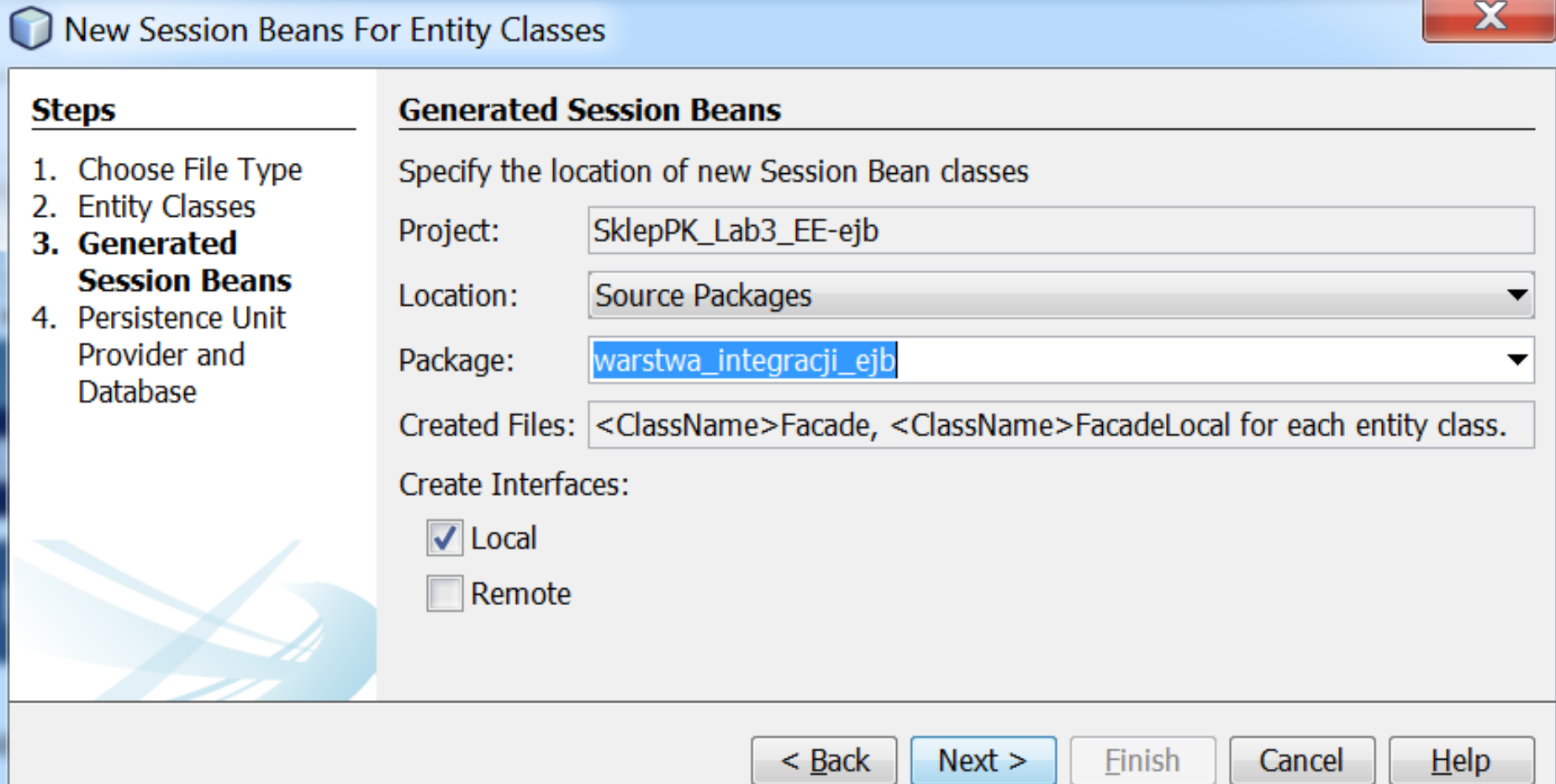
warstwa_biznesowa.entity.Produkt1

Include Referenced Classes

Create Persistence Unit

< Back Next > Finish Cancel Help

Generowanie komponentu typu **Session Bean** do utrwalania obiektu typu **Produkt1** - wybór w **Create Interfaces: Local** jako zasięgu komponentu EJB typu Session oraz wpisanie nazwy pakietu w polu **Package**



New Session Beans For Entity Classes

Steps

1. Choose File Type
2. Entity Classes
- 3. Generated Session Beans**
4. Persistence Unit Provider and Database

Generated Session Beans

Specify the location of new Session Bean classes

Project: SklepPK_Lab3_EE-ejb

Location: Source Packages

Package: warstwa_integracji_ejb

Created Files: <ClassName>Facade, <ClassName>FacadeLocal for each entity class.

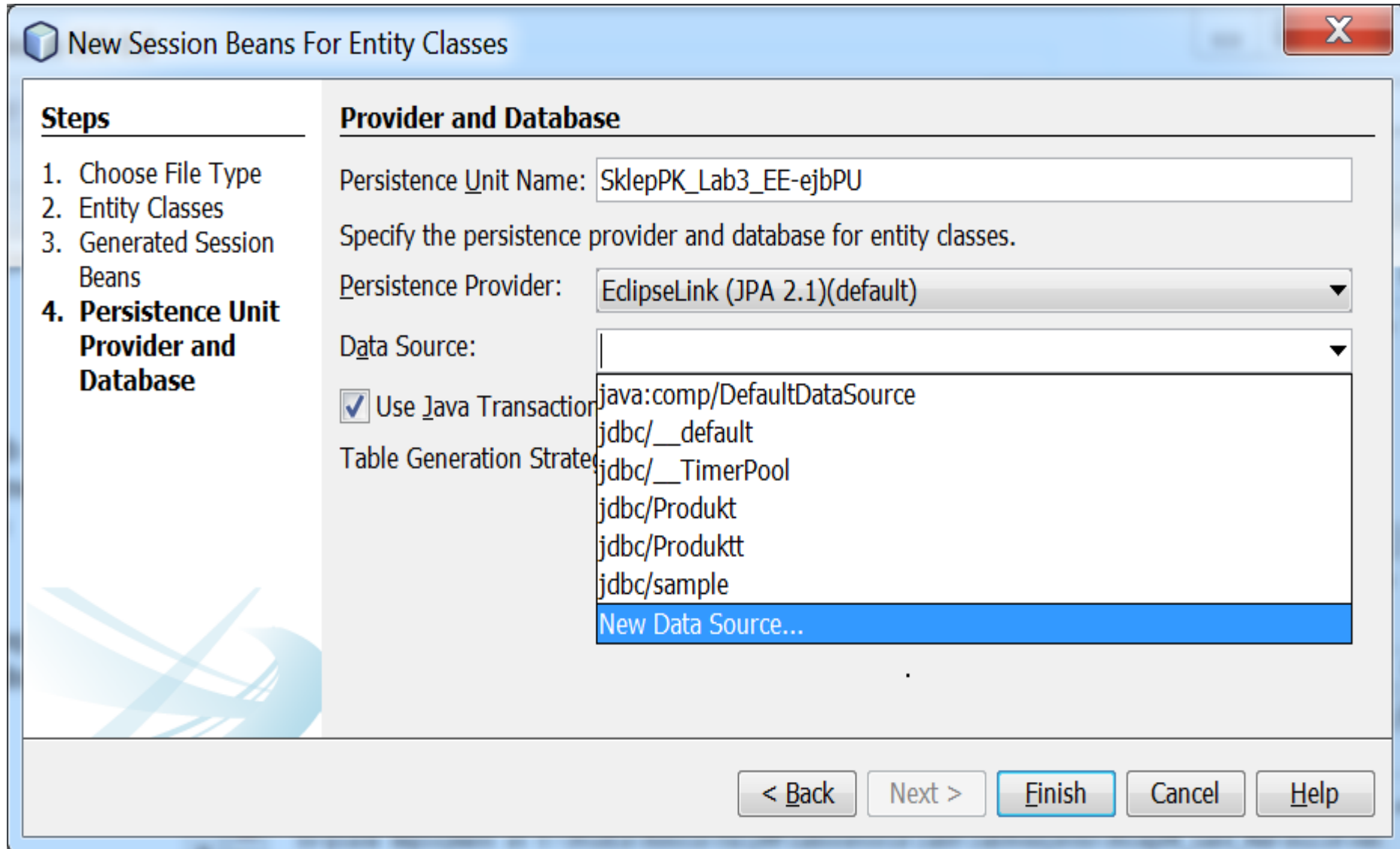
Create Interfaces:

Local

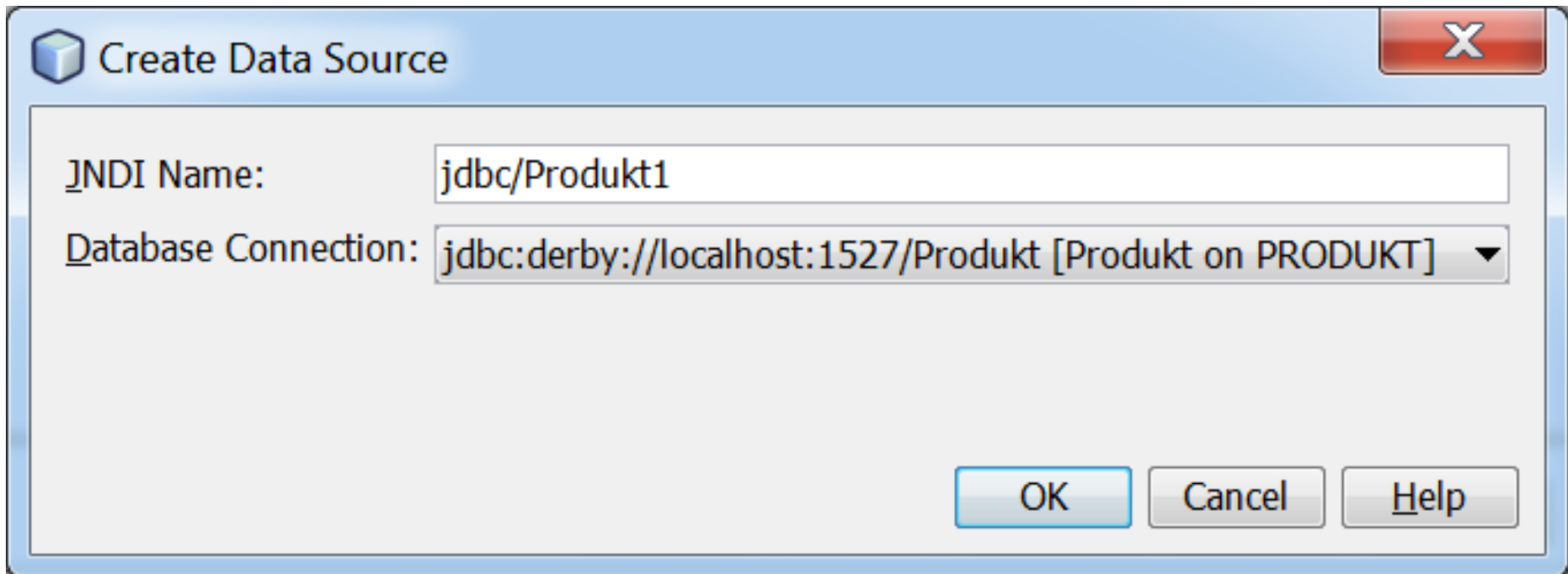
Remote

< Back Next > Finish Cancel Help

Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) - utworzenie połączenia z bazą danych (wykonaną w kroku **2**) za pomocą pozycji **New Data Source...**



Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) - wpisanie nazwy w polu **JNDI Name** (**Java Naming and Directory Interface – usługa do odkrywania i wyszukiwania danych oraz obiektów za pomocą nazw**) reprezentującego połączenie z bazą danych, wybraną z listy **Database Connection**



Create Data Source

JNDI Name:

Database Connection:

OK Cancel Help

Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) – rezultat, który zostanie zmodyfikowany

New Session Beans For Entity Classes

Steps

1. Choose File Type
2. Entity Classes
3. Generated Session Beans
- 4. Persistence Unit Provider and Database**

Provider and Database

Persistence Unit Name: SklepPK_Lab3_EE-ejbPU

Specify the persistence provider and database for entity classes.

Persistence Provider: EclipseLink (JPA 2.1)(default) ▼

Data Source: java:module/jdbc/Produkt1 ▼

Use Java Transaction APIs

Table Generation Strategy: Create Drop and Create None

< Back

Next >

Finish

Cancel

Help

Widok komponentu **Session Bean** fo **Entity Classes** (JPA)

Produkt1Facade, który implementuje interfejs **Produkt1FacadeLocal** i dziedziczy po klasie abstrakcyjnej **AbstractFacade**

SklepPK_Lab3_EE-ejb - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

- SklepPK_Lab3_EE
 - Java EE Modules
 - SklepPK_Lab3_EE-ejb.jar
 - Configuration Files
 - Server Resources
 - SklepPK_Lab3_EE-ejb
 - Source Packages
 - warstwa_biznesowa_ejb
 - Fasada_warstwy_biznesowej_ejb.java
 - warstwa_integracji_ejb
 - AbstractFacade.java
 - Produkt1Facade.java
 - Test Packages
 - Libraries
 - Test Libraries
 - Enterprise Beans
 - Configuration Files
 - META-INF
 - glassfish-resources.xml
 - MANIFEST.MF
 - persistence.xml
 - Server Resources

Members

- Produkt1Facade :: AbstractFacade<Produkt1>
 - Produkt1Facade()

```
1  ...5 lines
6  package warstwa_integracji_ejb;
7  import javax.ejb.Stateless;
8  import javax.persistence.EntityManager;
9  import javax.persistence.PersistenceContext;
10 import warstwa_biznesowa.entity.Produkt1;
11 /**...4 lines */
15 @Stateless
16 public class Produkt1Facade extends AbstractFacade<Produkt1> {
17
18     @PersistenceContext(unitName = "SklepPK_Lab3_EE-ejbPU")
19     private EntityManager em;
20     @Override
21     protected EntityManager getEntityManager() {
22         return em;
23     }
24     public Produkt1Facade() {
25         super(Produkt1.class);
26     }
27 }
28
```

warstwa_integracji_ejb.Produkt1Facade > Produkt1Facade >

Output

- Java DB Database Process
- GlassFish Server 4.1.1
- SklepPK_Lab3_Web (run)

Widok wygenerowanego interfejsu **Produkt1FacadeLocal**

The screenshot displays the NetBeans IDE 8.2 interface. The main window title is "SklepPK_Lab3_EE-ejb - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigat, Sourc, Refacto, Run, Debuç, Profik, Tear, Tool, Window, and Help. A search bar is present with the text "Search (Ctrl+I)".

The left sidebar shows a project tree for "SklepPK_Lab3_EE-ejb". The "Source Packages" folder is expanded, showing "warstwa_biznesowa_ejb" and "warstwa_integracji_ejb". Under "warstwa_integracji_ejb", the files "AbstractFacade.java", "Produkt1Facade.java", and "Produkt1FacadeLocal.java" are listed. "Produkt1FacadeLocal.java" is selected.

The main editor window displays the source code of "Produkt1FacadeLocal.java". The code is as follows:

```
1  ...5 lines
6  package warstwa_integracji_ejb;
7
8  import java.util.List;
9  import javax.ejb.Local;
10 import warstwa_biznesowa.entity.Produkt1;
11
12 /**...4 lines */
16 @Local
17
18 public interface Produkt1FacadeLocal {
19     void create(Produkt1 produkt1);
20     void edit(Produkt1 produkt1);
21     void remove(Produkt1 produkt1);
22     Produkt1 find(Object id);
23     List<Produkt1> findAll();
24     List<Produkt1> findRange(int[] range);
25     int count();
26 }
```

The bottom of the IDE shows a "count - Navigator" window with a "Members" list containing "create(Produkt1 produkt1)". The "Output - Java DB Database Process" window shows the message: "Thu Apr 06 21:17:35 CEST 2017 : Security manager installed using the B". The system tray at the bottom right shows the time "24:17" and the text "INS".

Wygenerowana klasa abstrakcyjna **AbstractFacade** (część 1)

The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows the source code of the `AbstractFacade` class, which is part of the `warstwa_integracji_ejb` package. The code includes imports for `java.util.List` and `javax.persistence.EntityManager`, and a class definition for `AbstractFacade` with a generic type parameter `T`. The class has a private field `entityClass` and several methods: `AbstractFacade` constructor, `getEntityManager`, `create`, `edit`, and `remove`.

```
1 ...5 lines
6 package warstwa_integracji_ejb;
7 import java.util.List;
8 import javax.persistence.EntityManager;
9 /**...4 lines */
10 public abstract class AbstractFacade<T> {
11
12     private Class<T> entityClass;
13
14     public AbstractFacade(Class<T> entityClass) {
15         this.entityClass = entityClass;
16     }
17     protected abstract EntityManager getEntityManager();
18
19     public void create(T entity) {
20         getEntityManager().persist(entity);
21     }
22     public void edit(T entity) {
23         getEntityManager().merge(entity);
24     }
25     public void remove(T entity) {
26         getEntityManager().remove(getEntityManager().merge(entity));
27     }
28 }
29
30
```

The left sidebar shows the project structure for `SklepPK_Lab3_EE`, including source packages like `warstwa_biznesowa_ejb` and `warstwa_integracji_ejb`. The bottom status bar shows the current cursor position at 28:1 and the text "INS".

Wygenerowana klasa abstrakcyjna **AbstractFacade** (część 2)

The screenshot shows the NetBeans IDE 8.2 interface. The main editor displays the `AbstractFacade.java` file with the following code:

```
32 public T find(Object id) {
33     return getEntityManager().find(entityClass, id);
34 }
35 public List<T> findAll() {
36     javax.persistence.criteria.CriteriaQuery cq =
37         getEntityManager().getCriteriaBuilder().createQuery();
38     cq.select(cq.from(entityClass));
39     return getEntityManager().createQuery(cq).getResultList();
40 }
41 public List<T> findRange(int[] range) {
42     javax.persistence.criteria.CriteriaQuery cq =
43         getEntityManager().getCriteriaBuilder().createQuery();
44     cq.select(cq.from(entityClass));
45     javax.persistence.Query q = getEntityManager().createQuery(cq);
46     q.setMaxResults(range[1] - range[0] + 1);
47     q.setFirstResult(range[0]);
48     return q.getResultList();
49 }
50 public int count() {
51     javax.persistence.criteria.CriteriaQuery cq =
52         getEntityManager().getCriteriaBuilder().createQuery();
53     javax.persistence.criteria.Root<T> rt = cq.from(entityClass);
54     cq.select(getEntityManager().getCriteriaBuilder().count(rt));
55     javax.persistence.Query q = getEntityManager().createQuery(cq);
56     return ((Long) q.getSingleResult()).intValue();
57 }
```

The left sidebar shows the project structure for `SklepPK_Lab3_EE-ejb`, with `AbstractFacade.java` selected under `warstwa_integracji_ejb`. The bottom status bar shows the `Output` window with the following tabs: `Java DB Database Process`, `GlassFish Server 4.1.1`, and `SklepPK_Lab3_Web (run)`.

Przekazanie nadawania **id** klasie **Produkt1** typu **Entity** mechanizmom utrwalania JPA – **modyfikacja kodu metody utworz_produkt** do dodawania obiektu typu Produkt1 (Entity) w klasie **Fasada_warstwy_biznesowej** (projekt **Sklep_6SE_1**)

```
public void utworz_produkt(String dane[], Date data) {  
    Produkt1 produkt = new Produkt1();  
    // klucz++;  
    // produkt.setId(new Long(klucz));  
    produkt.setNazwa(dane[0]);  
    produkt.setCena(Float.parseFloat(dane[1]));  
    produkt.setPromocja(Integer.parseInt(dane[2]));  
    produkt.setData_produkcji(data);  
    dodaj_produkt(produkt);  
}
```

gdzie w klasie typu **Produkt1** atrybut: **strategy = GenerationType.AUTO** adnotacji **@GeneratedValue** oznacza możliwość generowania klucza głównego podczas utrwalania przez aplikację lub przez komponenty JPA:

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

Modyfikacja kodu metody `produkt_transfer` generującej model obiekt transferowy typu **Produkt_dto** – modyfikacja wynika z braku wartości **id**, która dopiero zostanie nadana po zapisie danego obiektu typu `Produkt_dto` do bazy danych (w klasie **Fasada_warstwy_biznesowej** - projekt **Sklep_6SE_1**)

```
public Produkt_dto produkt_transfer(Produkt1 produkt) {  
    Produkt_dto pom = new Produkt_dto();  
    pom.setId(produkt.getId());  
    pom.setNazwa(produkt.getNazwa());  
    pom.setCena(produkt.getCena());  
    pom.setPromocja(produkt.getPromocja());  
    pom.setData_produkcji(produkt.getData_produkcji());  
    pom.setCena_brutto(produkt.cena_brutto());  
    return pom;  
}
```

Przekazanie nadawania id klasie Produkt1 typu Entity mechanizmowi utrwalania JPA
– modyfikacja kodu metody **items()** w klasie **Fasada_warstwy_biznesowej** -
projekt **Sklep_6SE_1**. Dodanie kodu metody **getId()** w klasie **Produkt1**

```
public ArrayList<ArrayList<String>> items() {  
    ArrayList<ArrayList<String>> dane = new ArrayList();  
    for (Produkt1 p : produkty) {  
        ArrayList<String> wiersz = new ArrayList();  
        wiersz.add(p.getId_().toString());  
        wiersz.add(p.getNazwa());  
        wiersz.add("" + p.getCena());  
        wiersz.add("" + p.getPromocja());  
        wiersz.add(p.getData_produkcji().toString());  
        wiersz.add("" + p.cena_brutto());  
        dane.add(wiersz);  
    }  
    return dane;  
}
```

gdzie w metodzie **getId_()** w obiekcie typu **Produkt1**, przed zapisaniem do bazy danych, gdy **id** jest równe null:

```
public Long getId_() {  
    if(id==null)  
        return new Long(0);  
    return id;  
}
```

Dodana metoda **produkty_z_bazy_danych** w klasie **Fasada_warstwy_biznesowej** (projekt **Sklep_6SE_1**) do zapisu w aplikacji kolekcji **produkty** danych **produkty_** pobranych z bazy danych, przekazanych przez listę parametrów metody

```
public void produkty_z_bazy_danych (List<Produkt1> produkty_)  
{  
    produkty.clear();  
    produkty.addAll(produkty_);  
}
```

Utworzenie referencji do komponentu **Produkt1Facade** w klasie **Fasada_warstwy_biznesowej_ejb** – **Insert Code/Call Enterprise Bean** i wybór z listy ziarna typu **Produkt1Facade** – pojawiła się referencja typu **Produkt1FacadeLocal**

SklepPK_Lab3_EE-ejb - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

SklepPK_Lab3_EE
SklepPK_Lab3_EE-ejb
Source Packages
warstwa_biznesowa_ejb
Fasada_warstwy_biznesowej_ejb.java
warstwa_integracji_ejb
AbstractFacade.java
Produkt1Facade.java
Produkt1FacadeLocal.java
Test Packages
Libraries
Test Libraries
Enterprise Beans
Configuration Files
Server Resources
SklepPK_Lab3_Web
Web Pages
WEB-INF
resources
warstwa_internetowa_jsf
baza.xhtml
dodaj_produkt2.xhtml

produkt1Facade - Navigator

Members

<empty>

- setStan(boolean stan)
- utworz_produkt(Produkt_dto produkt_dto)
- fasada : Fasada_warstwy_biznesowej
- produkt1Facade : Produkt1FacadeLocal

```
1 ... 5 lines
6 package warstwa_biznesowa_ejb;
7
8 import java.util.ArrayList;
9 import javax.ejb.EJB;
10 import javax.ejb.Stateless;
11 import warstwa_biznesowa.Fasada_warstwy_biznesowej;
12 import warstwa_biznesowa.dto.Produkt_dto;
13 import warstwa_integracji_ejb.Produkt1FacadeLocal;
14
15 /**... 4 lines */
19 @Stateless
20 public class Fasada_warstwy_biznesowej_ejb
21     implements Fasada_warstwy_biznesowej_ejbRemote {
22
23     @EJB
24     private Produkt1FacadeLocal produkt1Facade;
25
26     Fasada_warstwy_biznesowej fasada = new Fasada_warstwy_biznesowej();
27
28     public void utworz_produkt(Produkt_dto produkt_dto) {
29         fasada.utworz_produkt(produkt_dto);
30     }
31 }
```

warstwa_biznesowa_ejb.Fasada_warstwy_biznesowej_ejb > produkt1Facade >

Output - Java DB Database Process

Thu Apr 06 21:17:35 CEST 2017 : Security manager installed using the Basic server security policy.

84

23:5

INS

Dodane metody implementujące w klasie **Fasada_wartwy_biznesowej_ejb** do utrwalania danych **zapisz()** i do pobierania danych **pobierz()** z bazy danych Produkt.

```
public void pobierz() {  
    List<Produkt1> pom = produkt1Facade.findAll();  
    fasada.produkty_z_bazy_danych(pom);  
}
```

```
public void zapisz() {  
    for (Produkt1 p : fasada.getProdukty()) {  
        Long id = p.getId();  
        if (id == null || produkt1Facade.find(p.getId()) == null) {  
            produkt1Facade.create(p);  
        }  
    }  
}
```

Dodana metoda `init()` w klasie **`Fasada_warstwy_biznesowej_ejb`**
(komponentu typu Session -Stateless)
do pobrania danych z bazy danych przy uruchomieniu tego ziarna –
adnotacja **`@PostConstruct`** dodana do metody **`init()`** powoduje
automatyczne wywołanie tej metody zawsze podczas tworzenia komponentu
`Fasada_warstwy_biznesowej_ejb`

`@PostConstruct`

```
public void init() {
```

```
    try {
```

```
        // System.out.println("update1");
```

```
        pobierz() ;
```

```
    } catch (Exception e) { }
```

```
}
```

Dodanie dwóch metod: **zapisz()**, **pobierz()** w interfejsie metod logiki biznesowej **Fasada_warstwy_biznesowej_ejbRemote** implementowane w kompomemencie EJB **Fasada_warstwy_biznesowej_ejb** z modułu EJB (**SklepPK_Lab3_EE-ejb**)

```
package warstwa_biznesowa_ejb;
import java.util.ArrayList;
import javax.ejb.Remote;
import warstwa_biznesowa.dto.Produkt_dto;
@Remote
public interface Fasada_warstwy_biznesowej_ejbRemote {
    public void utworz_produkt(Produkt_dto produkt_dto);
    public Produkt_dto dane_produktu();
    public ArrayList<ArrayList<String>> items();
    public ArrayList<Produkt_dto> items_();
    public int count();
    public ArrayList<Produkt_dto> findRange(int[] range);
    public boolean isStan();
    public void setStan(boolean stan);
    public boolean edit(Produkt_dto o_przed, Produkt_dto o_update);
    public void remove(Produkt_dto p);
    public void zapisz();
    public void pobierz();
}
```

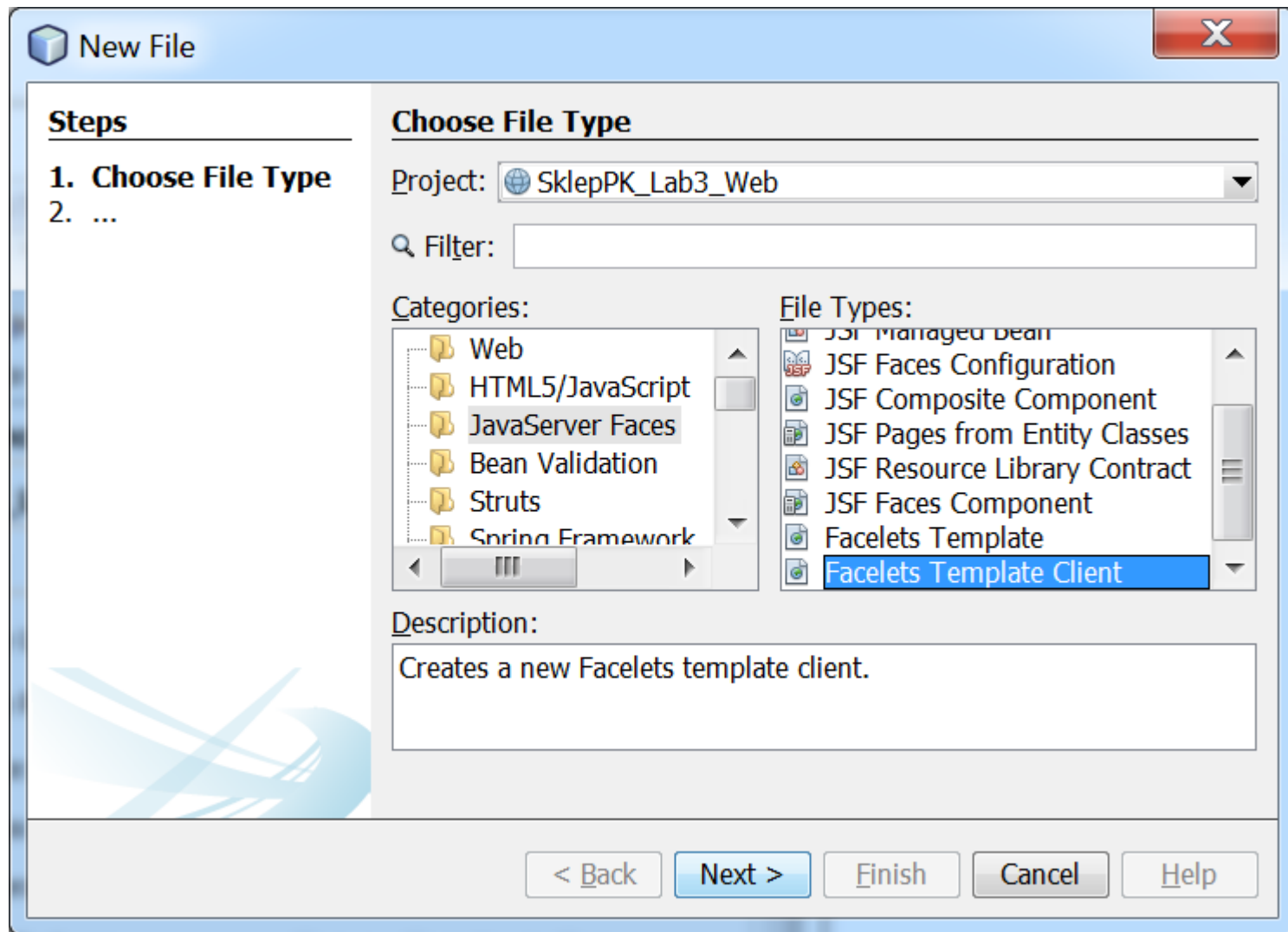
**Dodane metody `zapisz()`, `pobierz()` w klasie `Managed_produk`,
wywołujące metody `zapisz()`, `pobierz()` z komponentu EJB**

```
public String zapisz() {  
    fasada.zapisz();  
    return "/faces/index1";  
}
```

```
public String pobierz() {  
    fasada.pobierz();  
    refresh();  
    return "/faces/index1";  
}
```

Aktualzacja liczby stron po odczytaniu danych z bazy danych i aktualizacja modelu tabeli.

Dodanie strony **baza.xhtml** opartej na szablonie **template.xml**



Zawartość wygenerowanego pliku [baza.xhtml](#)

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:h="http://xmlns.jcp.org/jsf/html">
<body>
  <ui:composition template=" ../template.xhtml">
    <ui:define name="title">
      <h:outputText value="#{bundle['baza.tytul']}"></h:outputText>
    </ui:define>
```

Modyfikacja zawartości strony **baza.xhtml** opartej na szablonie **template.xml**

```
<ui:define name="content">
```

```
<h:form>
```

```
<h:commandLink action="#{managed_produkt.zapisz}"  
                value="#{bundle['baza.zapisz']}" /><br/>
```

```
<h:commandLink action="#{managed_produkt.pobierz}"  
                value="#{bundle['baza.pobierz']}" /><br/>
```

```
</h:form>
```

```
</ui:define>
```

```
</ui:composition>
```

```
</body>
```

```
</html>
```

Pierwszy comandLink obsługuje **zapis** do bazy danych, a drugi **odczyt** z bazy danych

Uzupełnienie kodu pliku **template.xhtml** – bloku "left", czyli dodanie znacznika **h:link** do wywoływania strony **baza.xhtml bundle**

```
<div id="left">  
    <h:link outcome="/faces/jsf/dodaj_produkt2"  
        value="Dodaj produkt"/> <br/>  
    <h:link outcome="/faces/jsf/lista_produktow"  
        value="Lista produktow"/><br/>  
    <h:link outcome="/faces/warstwa_internetowa_jsf/baza"  
        value="#{bundle['baza.tytul']}" />  
</div>
```

Baza danych jest pusta przed uruchomieniem aplikacji za pomocą **Deploy**

The screenshot shows the NetBeans IDE 8.2 interface. The title bar reads "SklepPK_Lab3_Web - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains icons for file operations and execution. The left sidebar shows a project tree with "Databases" expanded to "jdbc:derby://localhost:1527/Produkt [Produkt on PRODUKT]". Under "PRODUKT", there are folders for "Tables", "Views", and "Procedures". Below this is "jdbc:derby://localhost:1527/sample [app on APP]". The main editor displays the source code for "Managed_produk...". The code shows two methods: "zapisz()" and "pobierz()", both returning "/faces/index1". The "zapisz()" method is at lines 300-303, and "pobierz()" is at lines 306-308. The bottom status bar shows "305:1" and "INS".

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

Projects Files Services

Databases

- Java DB
- Drivers
- jdbc:derby://localhost:1527/Produkt [Produkt on PRODUKT]
 - PRODUKT**
 - Tables
 - Views
 - Procedures
 - Other schemas
- jdbc:derby://localhost:1527/sample [app on APP]
- Web Services
- Servers

Managed_produk... Fasad...

Source History

```
299
300     public String zapisz() {
301         fasada.zapisz();
302         return "/faces/index1";
303     }
304
305     public String pobierz() {
306         fasada.pobierz();
307         return "/faces/index1";
308     }
309 }
```

warstwa_internetowa.Managed_produk...

zapisz - Navigator

Members <empty>

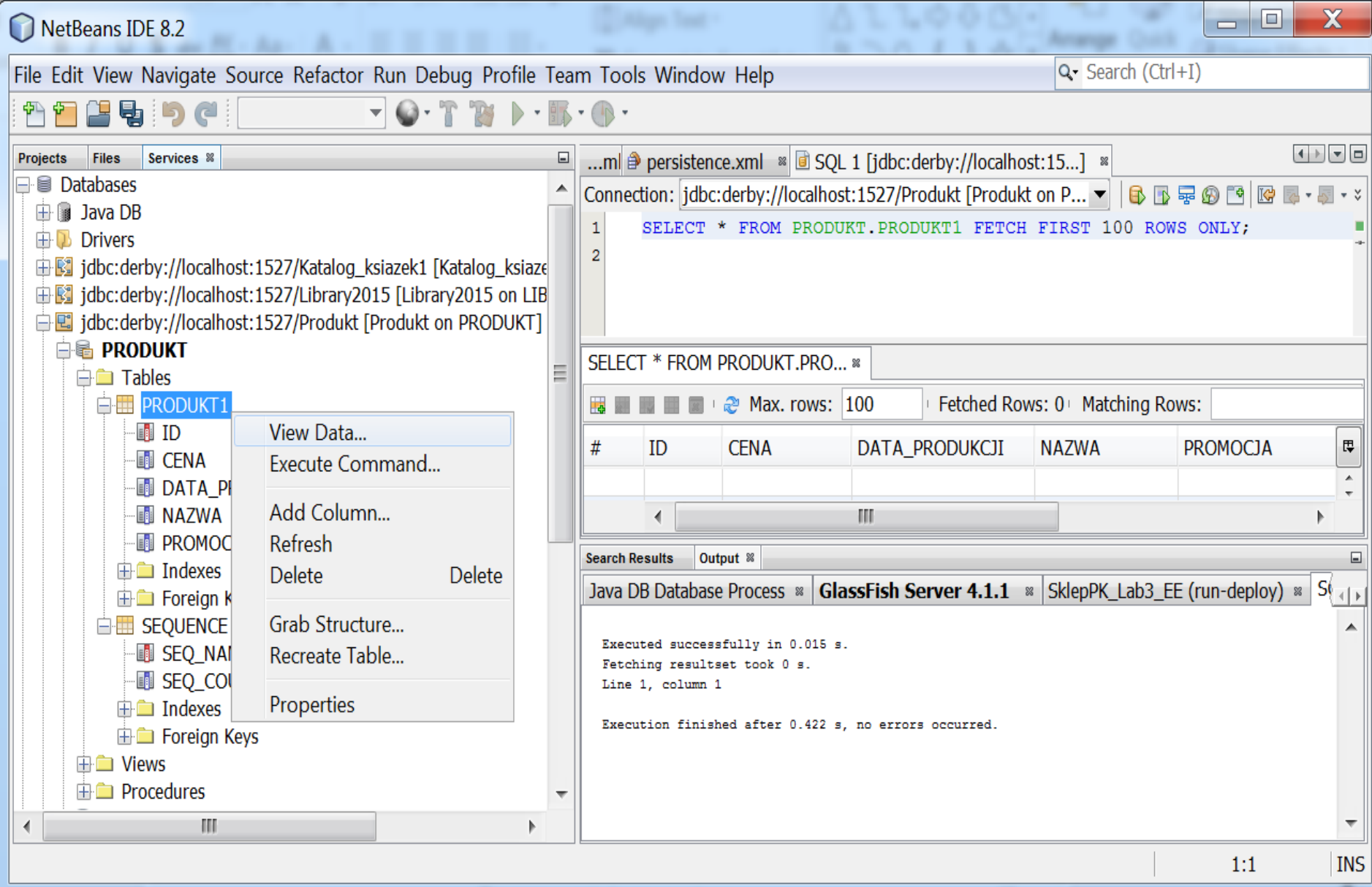
zakrespromocji(FacesContext context, UIComponent toValida

Output - Java DB Database Process

Thu Apr 06 21:17:35 CEST 2017 : Security manager instal:

305:1 INS

Po wykonaniu operacji **Deploy** w projekcie **SklepPK_Lab3_EE** - w bazie danych pojawią puste tabele: **Produkt1** i **Sequence** (widok po kliknięciu prawym klawiszem myszy na nazwę tabeli Produkt1 i wybór pozycji **View data...**)



NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

Databases

- Java DB
- Drivers
- jdbc:derby://localhost:1527/Katalog_ksiazek1 [Katalog_ksiazek1 on P...]
- jdbc:derby://localhost:1527/Library2015 [Library2015 on LIB...]
- jdbc:derby://localhost:1527/Produkt [Produkt on PRODUKT]

PRODUKT

- Tables
 - PRODUKT1**
 - ID
 - CENA
 - DATA_PROD...
 - NAZWA
 - PROMOCJA
- Indexes
- Foreign Keys
- SEQUENCE
- SEQ_NAME
- SEQ_COMMENT
- Indexes
- Foreign Keys
- Views
- Procedures

...ml persistence.xml SQL 1 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/Produkt [Produkt on P...]

```
1 SELECT * FROM PRODUKT.PRODUKT1 FETCH FIRST 100 ROWS ONLY;
```

2

SELECT * FROM PRODUKT.PRO... *

Max. rows: 100 | Fetched Rows: 0 | Matching Rows:

#	ID	CENA	DATA_PRODUKCJI	NAZWA	PROMOCJA

Search Results Output

Java DB Database Process GlassFish Server 4.1.1 SklepPK_Lab3_EE (run-deploy)

```
Executed successfully in 0.015 s.  
Fetching resultset took 0 s.  
Line 1, column 1  
  
Execution finished after 0.422 s, no errors occurred.
```

1:1 INS

Tablica **SEQUENCE** zawiera wartość wykorzystywaną przy generowaniu klucza głównego, gdy adnotacja przy id w klasie Produkt1 jest:

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

The screenshot shows the NetBeans IDE 8.2 interface. On the left, the 'Services' tab is active, displaying a tree view of the 'PRODUKT' database schema. The 'SEQUENCE' table is highlighted. The main editor shows a SQL query: `SELECT * FROM PRODUKT."SEQUENCE" FETCH FIRST 100 ROWS ONLY;`. Below the query, the results are displayed in a table with the following data:

#	SEQ_NAME	SEQ_COUNT
1	SEQ_GEN	50

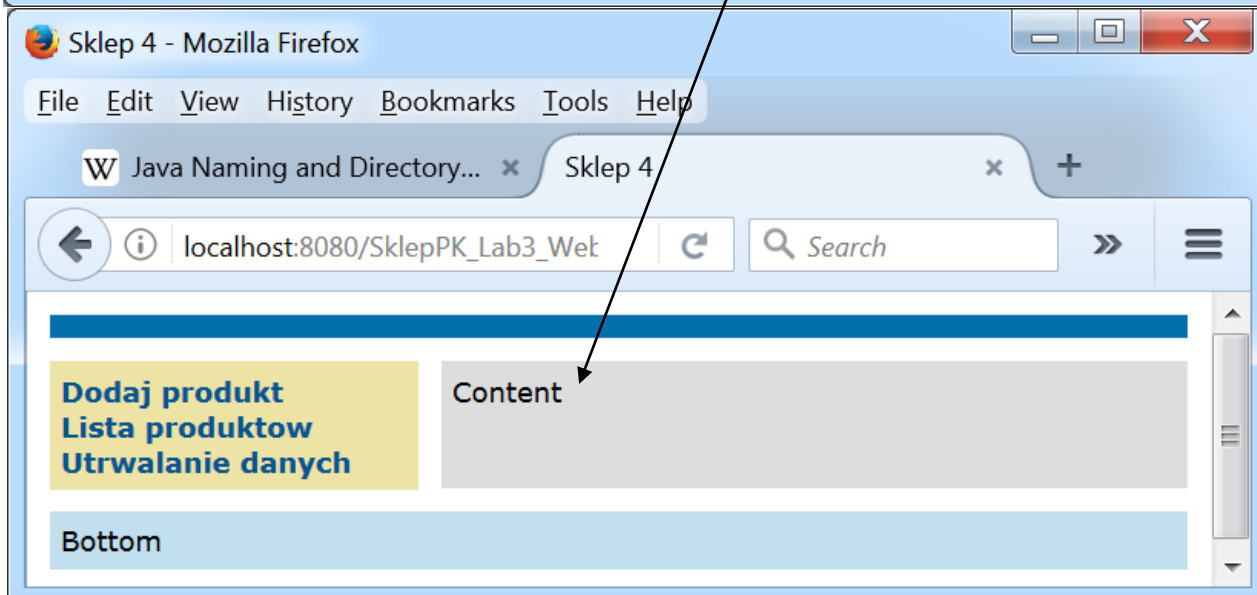
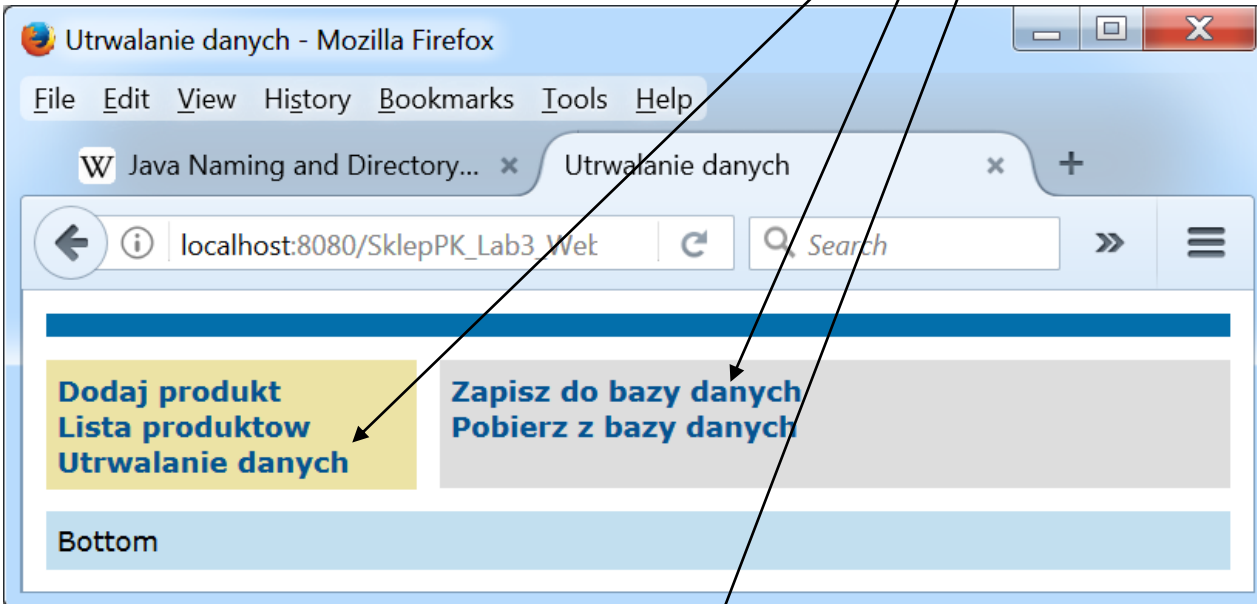
The 'Output' window at the bottom shows the execution log, including 'Sklep_6JPA (run)' and 'SQL 1 execution'.

Uruchomienie programu

The image displays a web browser window with the following details:

- Window Title:** MenuDemo
- Menu:** A Menu, Inne Menu
- Browser Tabs:** Java Naming and Directory..., Sklep 4
- Address Bar:** localhost:8080/SklepPK_Lab3_Web/
- Page Content:**
 - Left sidebar (yellow background):
 - Dodaj produkt**
 - Lista produktów**
 - Utrwalanie danych**
 - Main content area (grey background): Content
 - Bottom bar (light blue background): Bottom
- Second Browser Window:** Shows the same page content as the first window, with the address bar displaying http://localhost:8080/SklepPK_Lab3_v and the tab title Sklep 4.

Zapis do bazy danych



Dane zapisane do bazy danych **Produkt** (wybór pozycji **View Data...**, klikając na nazwę tabeli **Produkt1**)

The screenshot shows the NetBeans IDE 8.2 interface. On the left, the 'Projects' pane shows a tree view of databases, with 'PRODUKT' expanded to show 'PRODUKT1' selected. The main editor area displays a SQL query: `SELECT * FROM PRODUKT.PRODUKT1 FETCH FIRST 100 ROWS ONLY;`. Below the query, a table of results is shown with columns: #, ID, CENA, DATA_PRODUKCJI, NAZWA, and PROMOCJA. The results are:

#	ID	CENA	DATA_PRODUKCJI	NAZWA	PROMOCJA
2	1	120.0	2017-04-28	Produkt1	10
1	2	230.0	2017-04-28	Produkt2	20

At the bottom, the 'Output' pane shows the execution log for 'Sklep_GUIPK_lab3_EE_Desktop (run)', indicating successful execution.

Dodanie kolejnej danej w aplikacji desktopowej

MenuDemo

A Menu Inne Menu

Nazwa
Produkt3

Cena
345

Promocja
30

Data
28-04-2017

Dodaj produkt

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt1	120.0	10	Fri Apr 28 02:00:00 C...	108.0
2	Produkt2	230.0	20	Fri Apr 28 02:00:00 C...	184.0
0	Produkt3	345.0	30	Fri Apr 28 00:58:25 C...	241.5

Produkty

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	10 %	piątek, 28-04-2017	108 zł	Rezultat Edycja Usun
2	Produkt2	230 zł	20 %	piątek, 28-04-2017	184 zł	Rezultat Edycja Usun
0	Produkt3	345 zł	30 %	czwartek, 27-04-2017	241,5 zł	Rezultat Edycja Usun

[Powrot](#) [Odśwież stronę](#)

Bottom

Wyświetlenie danych przechowywanych w aplikacji oraz po pobraniu z bazy danych

1 ..3 /3

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	10 %	piątek, 28-04-2017	108 zł	Rezultat Edycja Usun
2	Produkt2	230 zł	20 %	piątek, 28-04-2017	184 zł	Rezultat Edycja Usun
0	Produkt3	345 zł	30 %	czwartek, 27-04-2017	241,5 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

Dodaj produkt
Lista produktow
Utrwalanie danych

Zapisz do bazy danych
Pobierz z bazy danych

Bottom

Dodaj produkt
Lista produktow
Utrwalanie danych

Content

Bottom

1 ..2 /2

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
2	Produkt2	230 zł	20 %	piątek, 28-04-2017	184 zł	Rezultat Edycja Usun
1	Produkt1	120 zł	10 %	piątek, 28-04-2017	108 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

Po ponownym uruchomieniu aplikacji automatycznie zostaną wczytane dane z bazy danych do aplikacji spowodowane metodą z adnotacją **@PostConstruct** (p.5.4) w klasie **Fasada_warstwy_biznesowej_ejb**.

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Java Naming and Directory... x Lista produktow x Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

1 ..2 /2

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
2	Produkt2	230 zł	20 %	czwartek, 27-04-2017	184 zł	Rezultat Edycja Usun
1	Produkt1	120 zł	10 %	czwartek, 27-04-2017	108 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

Projects Files Services

- JDK 1.8 (Default)
- GlassFish Server 4.1.1
- Test Libraries
- Enterprise Beans
- Configuration Files
- Server Resources
- SklepPK_Lab3_Web
- Web Pages
- WEB-INF
- resources
- warstwa internetowa i

Navigator

Members <empty>

- Managed_produk
- Managed_produk()
- dane_produktu()
- destroy() : String
- dodaj_produk()
- getCena() : float
- getCena_brutto() : float

...ml PaginationHelper.java web.xml Managed_produk.

Source History

pomoc.PaginationHelper > nextPage > if (isHasNextPage()) >

Search Results Output

Java DB Database Process GlassFish Server 4.1.1 SklepPK

```

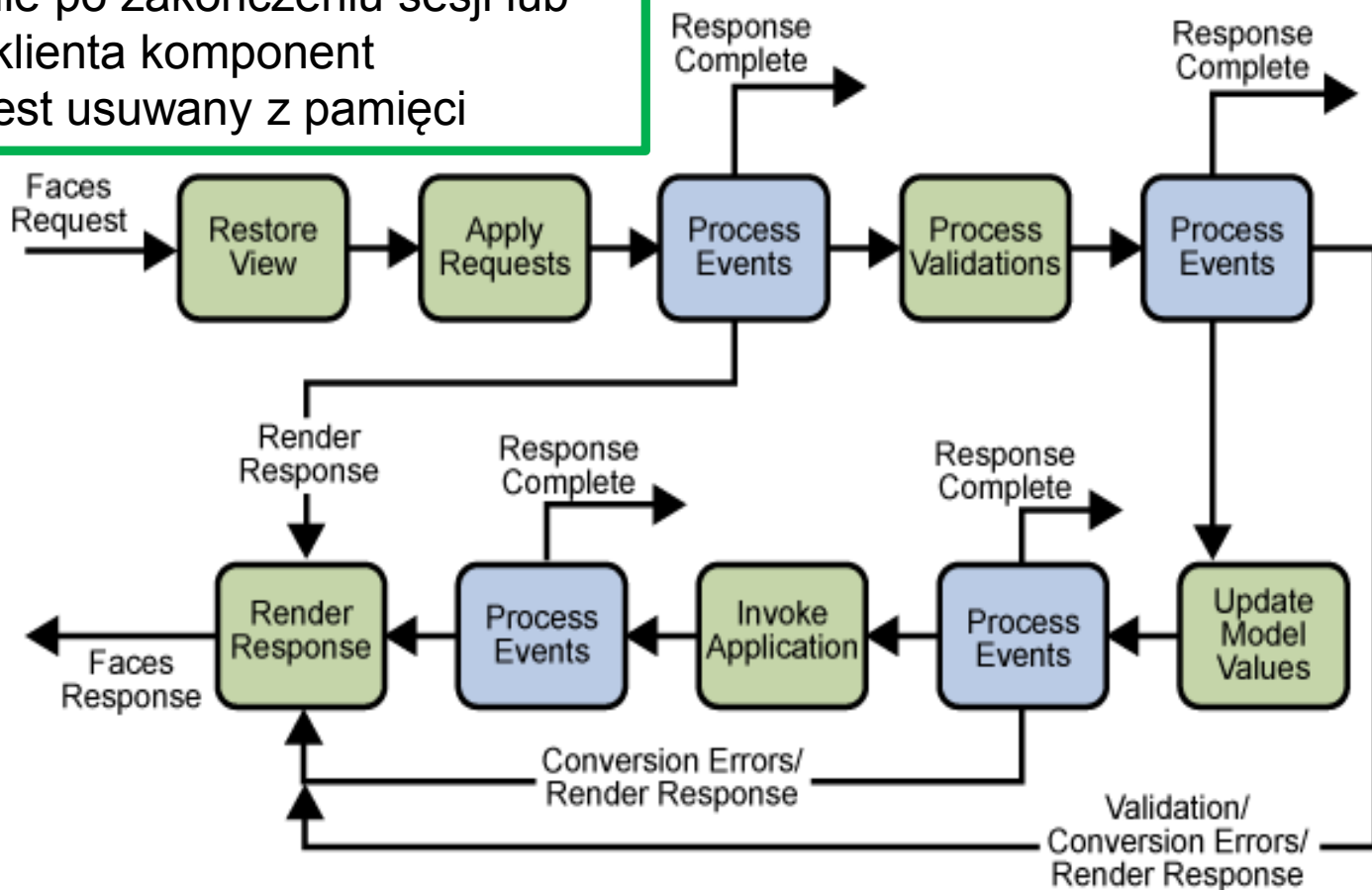
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] private org
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] org.glassfi
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] public org.
Info: Initializing Mojarra 2.2.12 ( 20150720-0848 https://svn.java.net
Info: Monitoring jndi:/server/SklepPK_Lab3_Web/WEB-INF/faces-config.xml
Info: Loading application [SklepPK_Lab3_Web] at [/SklepPK_Lab3_Web]
Info: SklepPK_Lab3_Web was successfully deployed in 735 milliseconds.
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_x
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_x
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_x
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_x
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
Info: update1

```

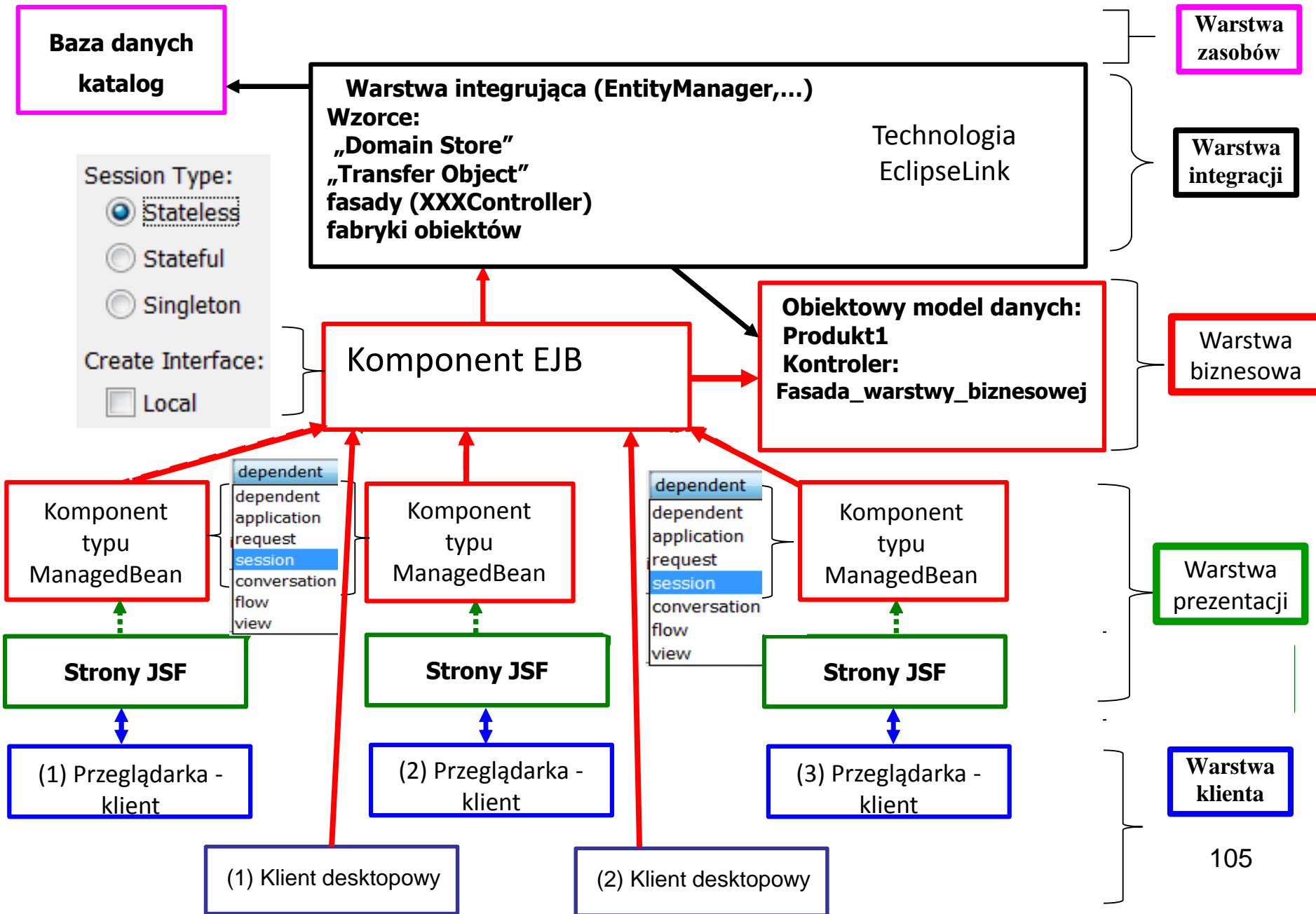
2. Wpływ czasu życia komponentu typu **Managed Bean** na obsługę stronicowania

Standard cyklu życia „Request-Response” dla JavaServer Faces

SessionScope: podczas uruchomienia instancji aplikacji klienta tworzony jest nowy komponent typu **Managed_produk**t – wszystkie jego dane są zaktualizowane podczas fazy Invoke Application. Po zakończeniu fazy Render Response komponent pozostaje w pamięci. Jeśli nie wystąpią błędy, jedynie po zakończeniu sesji lub zamknięciu aplikacji klienta komponent **Managed_produk**t jest usuwany z pamięci



Architektura aplikacji pięciowarstwowej -Java EE 7.0 JavaServer Faces



1. (1). Session Bean: rodzaj **Stateless**, Managed Bean: **SessionScoped**

SklepPK_Lab3_Web - NetBeans IDE 8.2

File Edit View Navigat Sourc Refacto Rur Debuç Profik Tean Tool: Window Help Search (Ctrl+I)

Projects Files Services

- index1.xhtml
- template.xhtml
- Source Packages
 - <default package>
 - pomoc
 - warstwa_internetowa
 - Managed_produkkt.java
- Test Packages
- Libraries
 - Sklep_6SE_1 - dist/Sklep

next - Navigator Members

- getRomanowicz() : int
- getStan() : int
- getZmiana() : int
- getZmiana1() : Zmiana_d
- getZmiana2() : Zmiana_d
- next() : String

Source History

```
22 import warstwa_biznesowa.dto.Produkt_dto;
23 import warstwa_biznesowa_ejb.Fasada_warstwy_bi
24
25 @Named(value = "managed_produkkt")
26 @SessionScoped
27 public class Managed_produkkt implements Action
28
29 @EJB
30 private Fasada_warstwy_biznesowej_ejbRemot
31
```

warstwa_internetowa.Managed_produkkt > next >

Search Results Output

Java DB Database Process GlassFish Server 4.1.1 SklepPK

```
Info: Loading application [SklepPK_Lab3_Web] at [/SklepPK_Lab3_Web]
Info: SklepPK_Lab3_Web was successfully deployed in 735 milliseconds.
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_f
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_y
```

28:1 INS

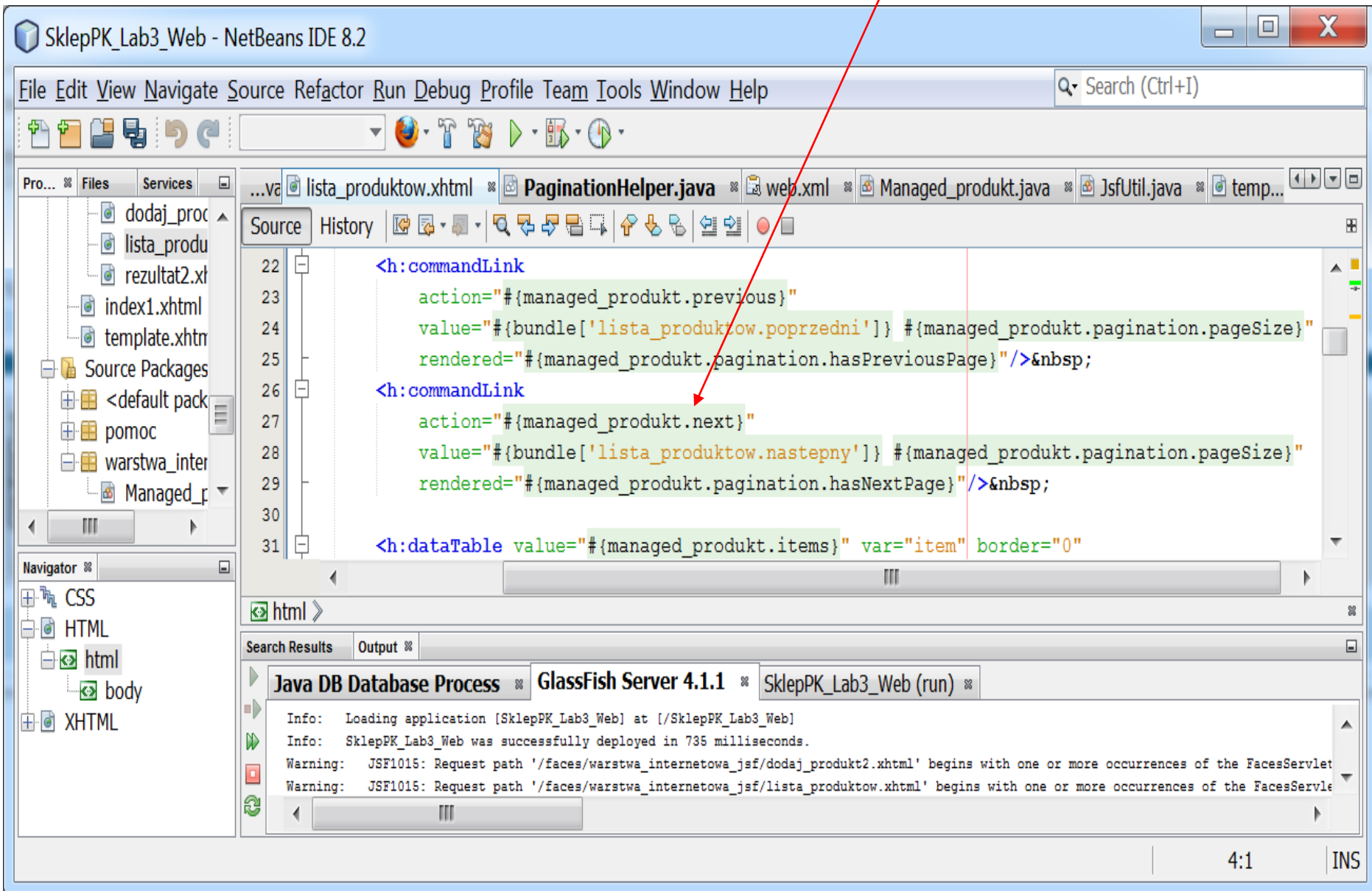
1. (2) Metoda **next()** do obsługi stronicowania w klasie typu Managed_produk - przystosowana do śledzenia stronicowania

The screenshot displays the NetBeans IDE 8.2 interface for the project "SklepPK_Lab3_Web". The main editor window shows the `PaginationHelper.java` file with the following code snippet:

```
44         return (page + 1) * pagesize + 1 <- getItemCount();
45     }
46
47     public void nextPage() {
48         System.out.println("page przed: "+page);
49         if (isHasNextPage()) {
50             page++;
51             System.out.println("page po: "+page);
52         }
53     }
```

The `nextPage()` method is highlighted in blue. The IDE also shows the project structure on the left, including files like `index1.xhtml` and `template.xhtml`, and the `Managed_produk.java` file. The bottom status bar shows the current page is 48:8 and the cursor is in Insert (INS) mode.

1. (3). Miejsce wywołania metody **next()** do obsługi stronicowania w klasie typu Managed_produk



The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows the source code of `Managed_produk.java`. The code is an XHTML snippet with the following content:

```
22 <h:commandLink
23     action="#{managed_produk.previous}"
24     value="#{bundle['lista_produk.poprzedni']} #{managed_produk.pagination.pageSize}"
25     rendered="#{managed_produk.pagination.hasPreviousPage}"/>&nbsp;
26 <h:commandLink
27     action="#{managed_produk.next}"
28     value="#{bundle['lista_produk.nastepny']} #{managed_produk.pagination.pageSize}"
29     rendered="#{managed_produk.pagination.hasNextPage}"/>&nbsp;
30
31 <h:dataTable value="#{managed_produk.items}" var="item" border="0"
```

A red arrow points from the title to the `action="#{managed_produk.next}"` attribute in the second `<h:commandLink>` element (line 27).

The interface also shows a project navigator on the left with a tree structure including `dodaj_produk`, `lista_produk`, `rezultat2.xf`, `index1.xhtml`, `template.xhtml`, `Source Packages`, `<default pack`, `pomoc`, `warstwa_inter`, and `Managed_p`. The bottom status bar shows the current page is 4:1 and the language is INS.

1. (4). Uruchomiona strona lista_produkow.xhtml – metoda **init** z adnotacją **@PostConstruct** pobrała dane z bazy danych do aplikacji

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml;jsess Search

Dodaj produkt
Lista produktow
Utrwalanie danych

1 ..3 /7 **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
3	Produkt3	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
1	Produkt1	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

1. (5). Przejście do strony drugiej po kliknięciu na przycisk **Następny 3**

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

4 ..6 /7 [Popzedni 3](#) [Nastepny 3](#)

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
6	Produkt6	2340 zł	20 %	czwartek, 27-04-2017	1 872 zł	Rezultat Edycja Usun
4	Produkt4	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

[Powrot](#) [Odswiez strone](#)

Bottom

1. (6). Widok pierwszej fazy Request po kliknięciu na przycisk **Następny 3**

The screenshot shows the NetBeans IDE 8.2 interface. The title bar reads "SklepPK_Lab3_Web - NetBeans IDE 8.2". The menu bar includes "File", "Edit", "View", "Navigat", "Sourc", "Refacto", "Rur", "Debuç", "Profik", "Team", "Tool:", "Window", and "Help". A search bar contains "Search (Ctrl+I)".

The left sidebar contains the "Projects" view with a tree structure:

- JDK 1.8 (Default)
- GlassFish Server 4.1.1
- Test Libraries
- Enterprise Beans
- Configuration Files
- Server Resources
- SklepPK_Lab3_Web (selected)
- Web Pages
- WEB-INF
- resources
- warstwa_internetowa

The "Navigator" view shows the "Members" of the "Managed_produk" class:

- Managed_produk() (diamond icon)
- dane_produkta() (circle icon)
- destroy() : String (circle icon)
- dodaj_produkta() (circle icon)
- getCena() : float (circle icon)

The main editor area shows the "Source" view of "PagingHelper.java" with the following breadcrumb: "pomoc.PagingHelper > nextPage > if (isHasNextPage())".

The "Output" window is active, displaying the following log messages:

```
Java DB Database Process    GlassFish Server 4.1.1    SklepPK_Lab3_Web
Info:   visiting unvisited references
WARN:   WELD-000411: Observer method [BackedAnnotatedMethod] private org.glassfish.
WARN:   WELD-000411: Observer method [BackedAnnotatedMethod] org.glassfi
WARN:   WELD-000411: Observer method [BackedAnnotatedMethod] public org.g
Info:   Initializing Mojarra 2.2.12 ( 20150720-0848 https://svn.java.net
Info:   Monitoring jndi:/server/SklepPK_Lab3_Web/WEB-INF/faces-config.xml
Info:   Loading application [SklepPK_Lab3_Web] at [/SklepPK_Lab3_Web]
Info:   SklepPK_Lab3_Web was successfully deployed in 735 milliseconds.
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza_produkta.jspx'
Info:   update1
Info:   page przed: 0
Info:   page po: 1
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produkta.jspx'
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza_produkta.jspx'
```

A red arrow points from the text "Widok pierwszej fazy Request" in the title to the "Info: page przed: 0" line in the output window.

The status bar at the bottom right shows "51:40" and "INS".

1. (7). Przejście do strony trzeciej po kliknięciu na przycisk **Następny 3**

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktow
Utrwalanie danych

7 ..7 /7 [Poprzedni 3](#)

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
51	Produkt8	880 zł	30 %	czwartek, 27-04-2017	616 zł	Rezultat Edycja Usun

[Powrot](#) [Odśwież stronę](#)

Bottom

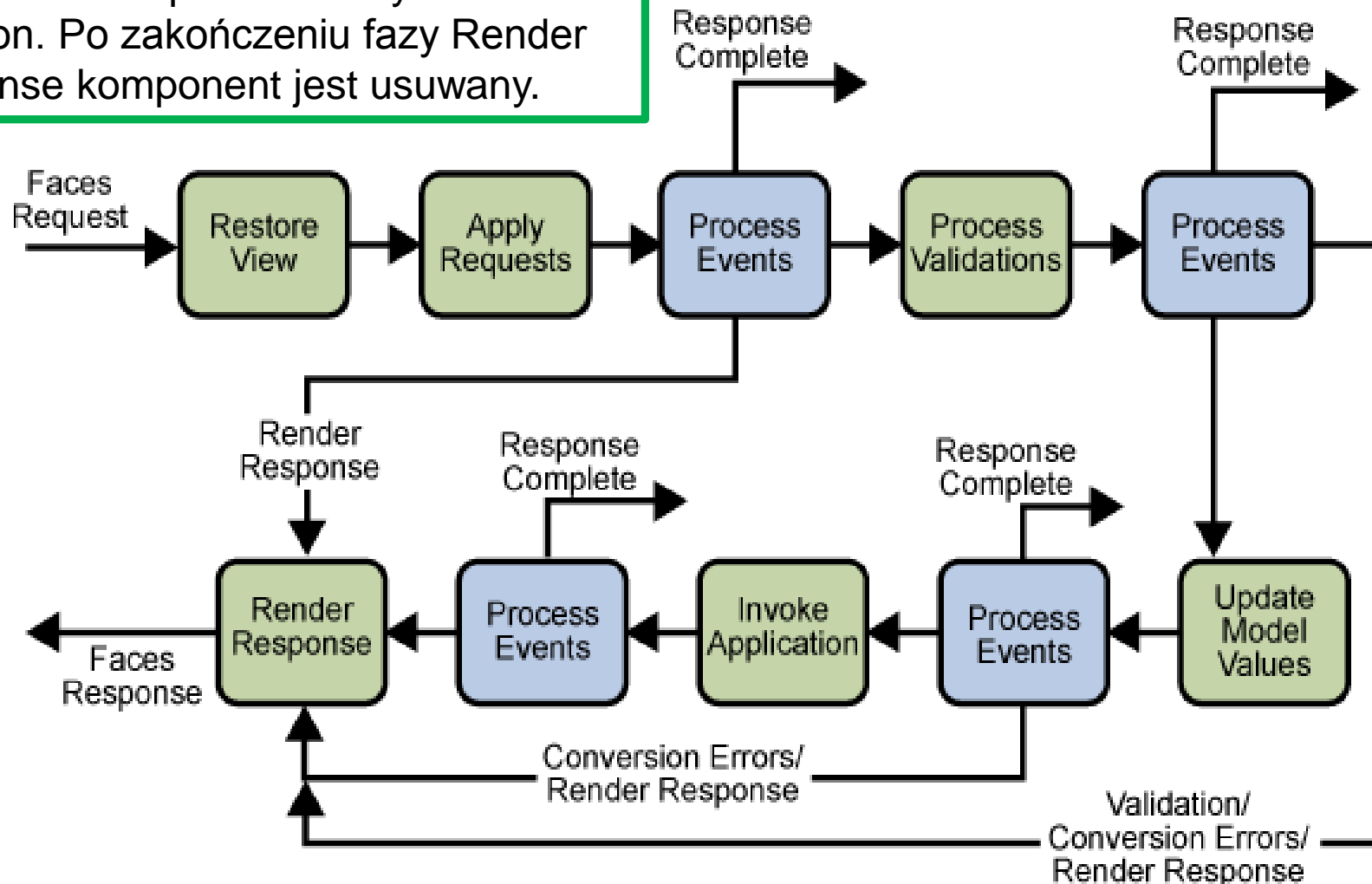
1. (8). Widok drugiej fazy Request po kliknięciu na przycisk **Następny 3**. Komponent typu **Managed_produk**t przechowuje w kolejnej faze Request numer bieżącej strony

The screenshot shows the NetBeans IDE 8.2 interface. The title bar reads "SklepPK_Lab3_Web - NetBeans IDE 8.2". The menu bar includes "File", "Edit", "View", "Navigat", "Source", "Refacto", "Rur", "Debug", "Profil", "Tear", "Tool", "Window", and "Help". The toolbar contains various icons for file operations and execution. The left sidebar shows the "Projects" view with a tree structure including "JDK 1.8 (Default)", "GlassFish Server 4.1.1", "Test Libraries", "Enterprise Beans", "Configuration Files", "Server Resources", "SklepPK_Lab3_Web", "Web Pages", "WEB-INF", and "resources". The "Navigator" view shows the "Managed_produk"t component with methods like "Managed_produk()", "dane_produk()", "destroy()", "dodaj_produk()", and "getCena()". The main editor area displays the "Managed_produk"t component with the "nextPage" method. The "Search Results" view shows the "Output" tab with the following log messages:

```
Info: Loading application [SklepPK_Lab3_Web] at [/SklepPK_Lab3_Web]
Info: SklepPK_Lab3_Web was successfully deployed in 735 milliseconds.
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
Info: update1
Info: page przed: 0
Info: page po: 1
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
Info: page przed: 1
Info: page po: 2
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_p
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xh
```

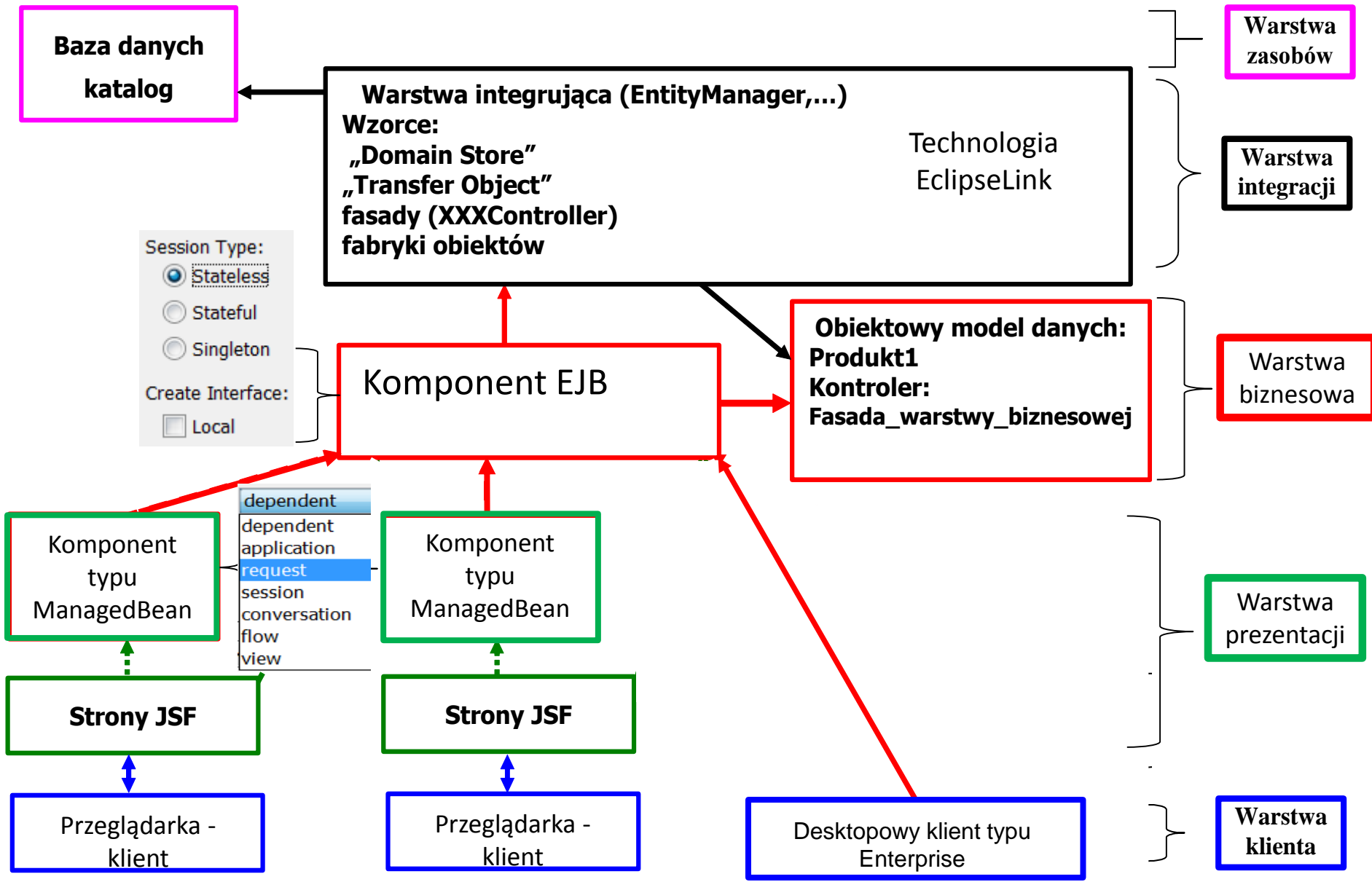
Standard cyklu życia „Request-Response” dla JavaServer Faces

RequestScope: podczas odebrania żądania tworzony jest nowy komponent typu **Managed_produk**t – wszystkie jego dane są zaktualizowane podczas fazy Invoke Application. Po zakończeniu fazy Render Response komponent jest usuwany.



Architektura aplikacji pięciowarstwowej -Java EE 7.0

JavaServer Faces



2. (1). Session Bean: rodzaj **Stateless**, Managed Bean: **RequestScoped**

The screenshot displays the NetBeans IDE 8.2 interface for a project named "SklepPK_Lab3_Web". The main editor window shows the source code for the `Managed_produk` class in `warstwa_internetowa.Managed_produk`. The code includes imports for `warstwa_biznesowa.dto.Produkt_dto` and `warstwa_biznesowa_ejb.Fasada_warstwy_biznesowej_ejbRemote`. The class is annotated with `@Named(value = "managed_produk")` and `@RequestScoped`, and implements the `ActionListener` interface. It is also annotated with `@EJB` and has a private field `Fasada_warstwy_biznesowej_ejbRemote fasada;`.

The left sidebar shows the project structure, including "Enterprise Beans", "Configuration Files", "Server Resources", and "Web Pages". The "Managed_produk - Navigator" window shows the class members, including `Managed_produk() :: ActionListener` and methods like `dane_produkту()`, `destroy() : String`, and `dodaj_produkту()`.

The bottom status bar shows the output window with the message: "BUILD SUCCESSFUL (total time: 3 seconds)".

```
22 import warstwa_biznesowa.dto.Produkt_dto;
23 import warstwa_biznesowa_ejb.Fasada_warstwy_biznesowej_ejbRemote;
24
25 @Named(value = "managed_produk")
26 @RequestScoped
27 public class Managed_produk implements ActionListener,
28
29     @EJB
30     private Fasada_warstwy_biznesowej_ejbRemote fasada;
31
```

2. (2) Uruchomiona strona lista_produkow.xhtml – metoda **init** z adnotacją **@PostConstruct** pobrała dane z bazy danych do aplikacji

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml;jse

1 ..3 /7 **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
3	Produkt3	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
1	Produkt1	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

2. (3). Przejście do strony drugiej po kliknięciu na przycisk **Następny 3**

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow x +

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml Search

Dodaj produkt
Lista produktow
Utrwalanie danych

4 ..6 /7 [Poprzedni 3](#) [Następny 3](#)

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
6	Produkt6	2340 zł	20 %	czwartek, 27-04-2017	1 872 zł	Rezultat Edycja Usun
4	Produkt4	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

2. (4). Widok pierwszej fazy Request po kliknięciu na przycisk **Następny 3**

SklepPK_Lab3_Web - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

Projects Files Services

- Libraries
 - Sklep_6SE_1 - dist/Sklep_6SE
 - SklepPK_interfejs_1 - dist/SklepPK_interfejs_1
- JDK 1.8 (Default)
- GlassFish Server 4.1.1
- Test Libraries
- Enterprise Beans
- Configuration Files
- Server Resources
- SklepPK_Lab3_Web

Managed_produk... web.xml Managed_produk.java JsfUtil.java tem...

Source History

```
22 import warstwa_biznesowa.dto.Produkt_dto;
```

warstwa_internetowa.Managed_produk

Search Results Output

Java DB Database Process GlassFish Server 4.1.1 SklepPK_Lab3_Web

```
Info: Initializing mojarra 2.2.14 ( 20130720-0040 https://svn.java.net/svn/mojarra)
Info: Monitoring jndi:/server/SklepPK_Lab3_Web/WEB-INF/faces-config.xml for
Info: Loading application [SklepPK_Lab3_Web] at [/SklepPK_Lab3_Web]
Info: SklepPK_Lab3_Web was successfully deployed in 846 milliseconds.
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
Info: update1
Info: page przed: 0
Info: page po: 1
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
```

Managed_produk - Navigator

Members <empty>

- Managed_produk :: ActionListener
 - Managed_produk()
 - dane_produkту()
 - destroy() : String
 - dodaj_produkту()
 - getCena() : float

2. (5). Nieudana próba przejścia do strony trzeciej po kliknięciu na przycisk **Następny 3**

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktow
Utrwalanie danych

4 ..6 /7 **Poprzedni 3** **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
6	Produkt6	2340 zł	20 %	czwartek, 27-04-2017	1 872 zł	Rezultat Edycja Usun
4	Produkt4	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

2. (6). Widok drugiej fazy Request po kliknięciu na przycisk **Następny 3** – widać, że **komponent typu Managed_produk**t jest tworzony ponownie i **zostanie utracona informacja o bieżącym numerze strony**

SklepPK_Lab3_Web - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

Projects Files Services

- Libraries
 - Sklep_6SE_1 - dist/Sklep_6SE
 - SklepPK_interfejs_1 - dist/SklepPK_interfejs_1
 - JDK 1.8 (Default)
 - GlassFish Server 4.1.1
- Test Libraries
- Enterprise Beans
- Configuration Files
- Server Resources
- SklepPK_Lab3_Web

Managed_produk - Navigator

Members <empty>

- Managed_produk :: ActionListener
 - Managed_produk()
 - dane_produkту()
 - destroy() : String
 - dodaj_produkту()
 - getCena() : float

Source History

```
22 import warstwa_biznesowa.dto.Produkt_dto;
```

warstwa_internetowa.Managed_produk

Search Results Output

Java DB Database Process GlassFish Server 4.1.1 SklepPK_Lab3_Web

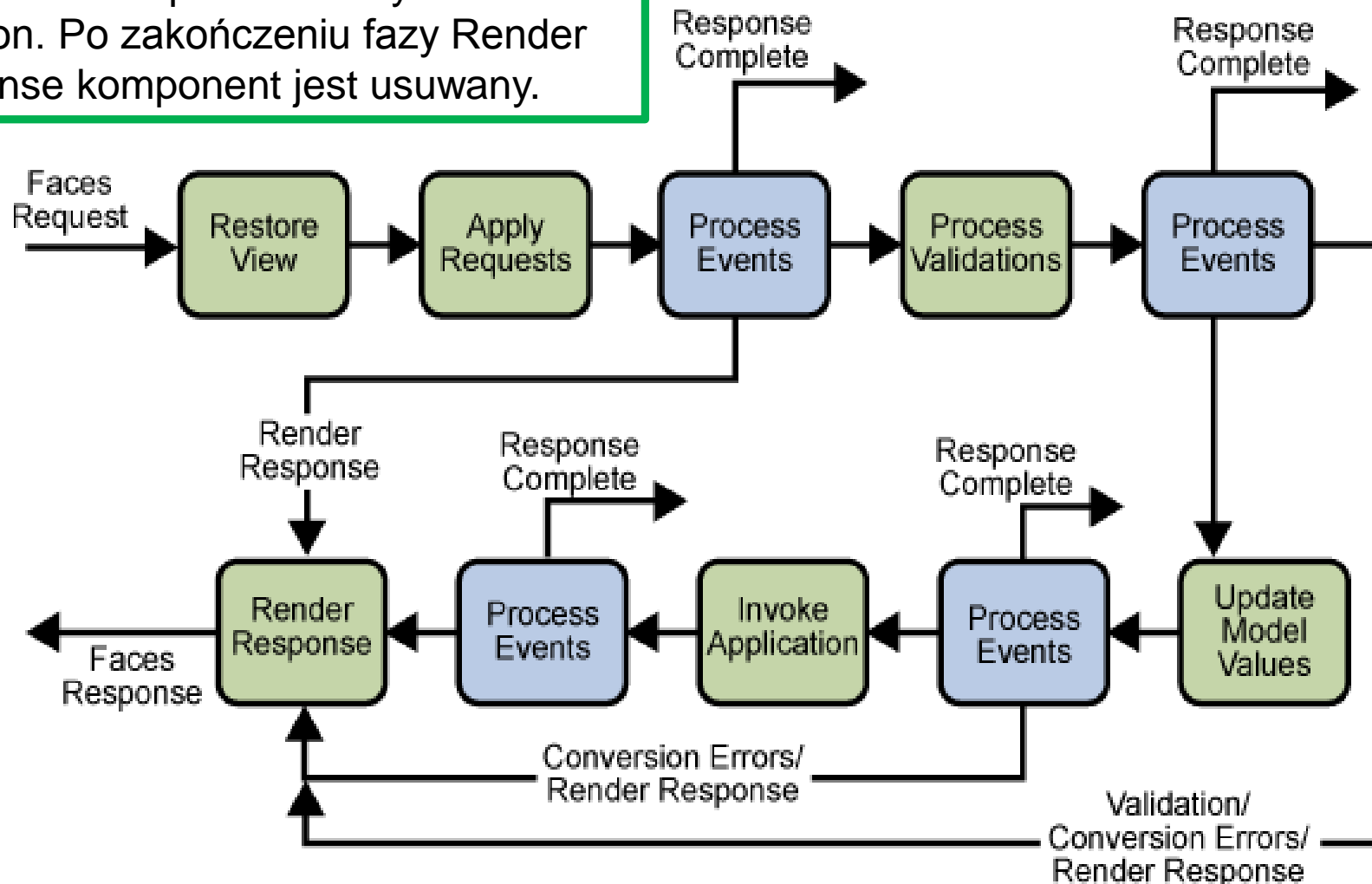
```
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
Info: update1
Info: page przed: 0
Info: page po: 1
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
Info: page przed: 0
Info: page po: 1
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/dodaj_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/lista_produk
Warning: JSF1015: Request path '/faces/warstwa_internetowa_jsf/baza.xhtml' b
```

26:9 INS

3. Wpływ komponentu typu **Managed Bean** (RequestScoped) na obsługę usuwania danych

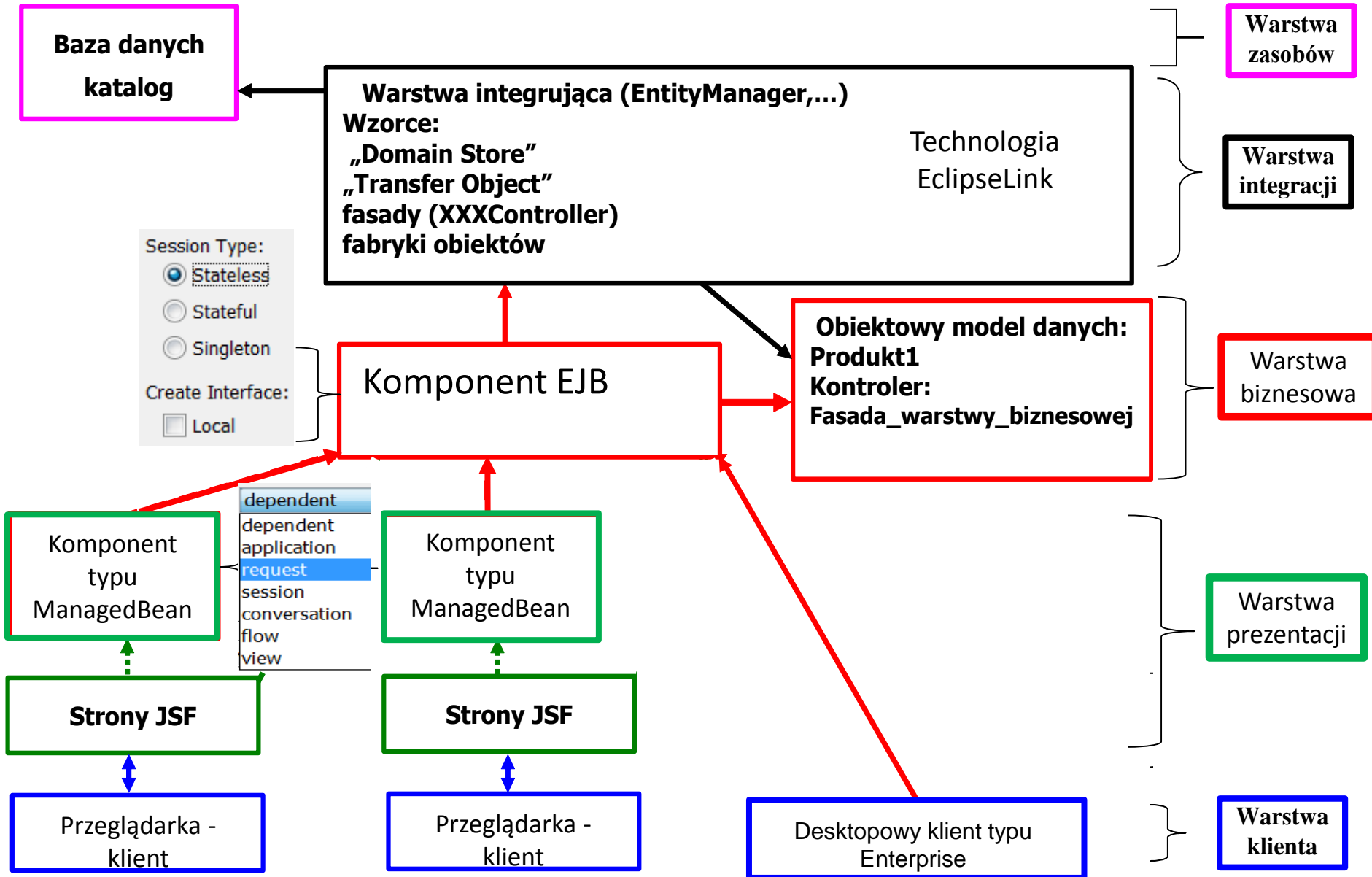
Standard cyklu życia „Request-Response” dla JavaServer Faces

RequestScope: podczas odebrania żądania tworzony jest nowy komponent typu **Managed_produk**t – wszystkie jego dane są zaktualizowane podczas fazy Invoke Application. Po zakończeniu fazy Render Response komponent jest usuwany.



Architektura aplikacji pięciowarstwowej -Java EE 7.0

JavaServer Faces



Nieudana próba usunięcia wiersza zawierającego Produkt5 – po usunięciu przejście na stronę pierwszą – został utracony numer strony

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktów
Utrwalanie danych

4 ..6 /7 **Poprzedni 3** **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
6	Produkt6	2340 zł	20 %	czwartek, 27-04-2017	1 872 zł	Rezultat Edycja Usun
4	Produkt4	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktów
Utrwalanie danych

1 ..3 /6 **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
3	Produkt3	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
1	Produkt1	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

• Produkt został usuniety

Bottom

Widok strony 2- w rzeczywistości usunięty został wiersz pierwszy z Produkt6 (skutek usunięcia) – jedynie poprawnie zainicjonana liczba stron. Wiersz ze strony trzeciej (z Produkt8) został przesunięty na stronę 2. Kolejna próba usunięcia – wybrany wiersz z Produkt8

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

4 ..6 /6 **Poprzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
4	Produkt4	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
51	Produkt8	880 zł	30 %	czwartek, 27-04-2017	616 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

Bottom

Nieudana próba usunięcia wiersza zawierającego Produkt8 – po usunięciu przejście na stronę pierwszą – został utracony numer strony. Jednak usunięty został wiersz z Produkt4

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktów
Utrwalanie danych

1 ..3 /6 **Następny 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
3	Produkt3	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
1	Produkt1	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
2	Produkt2	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun

Powrot Odśwież stronę

• Produkt został usunięty

Bottom

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów

localhost:8080/SklepPK_Lab3_Web/faces/warstwa_internetowa_jsf/lista_produkow.xhtml

Dodaj produkt
Lista produktów
Utrwalanie danych

4 ..5 /5 **Poprzedni 3**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
5	Produkt5	123 zł	12 %	czwartek, 27-04-2017	108,24 zł	Rezultat Edycja Usun
51	Produkt8	880 zł	30 %	czwartek, 27-04-2017	616 zł	Rezultat Edycja Usun

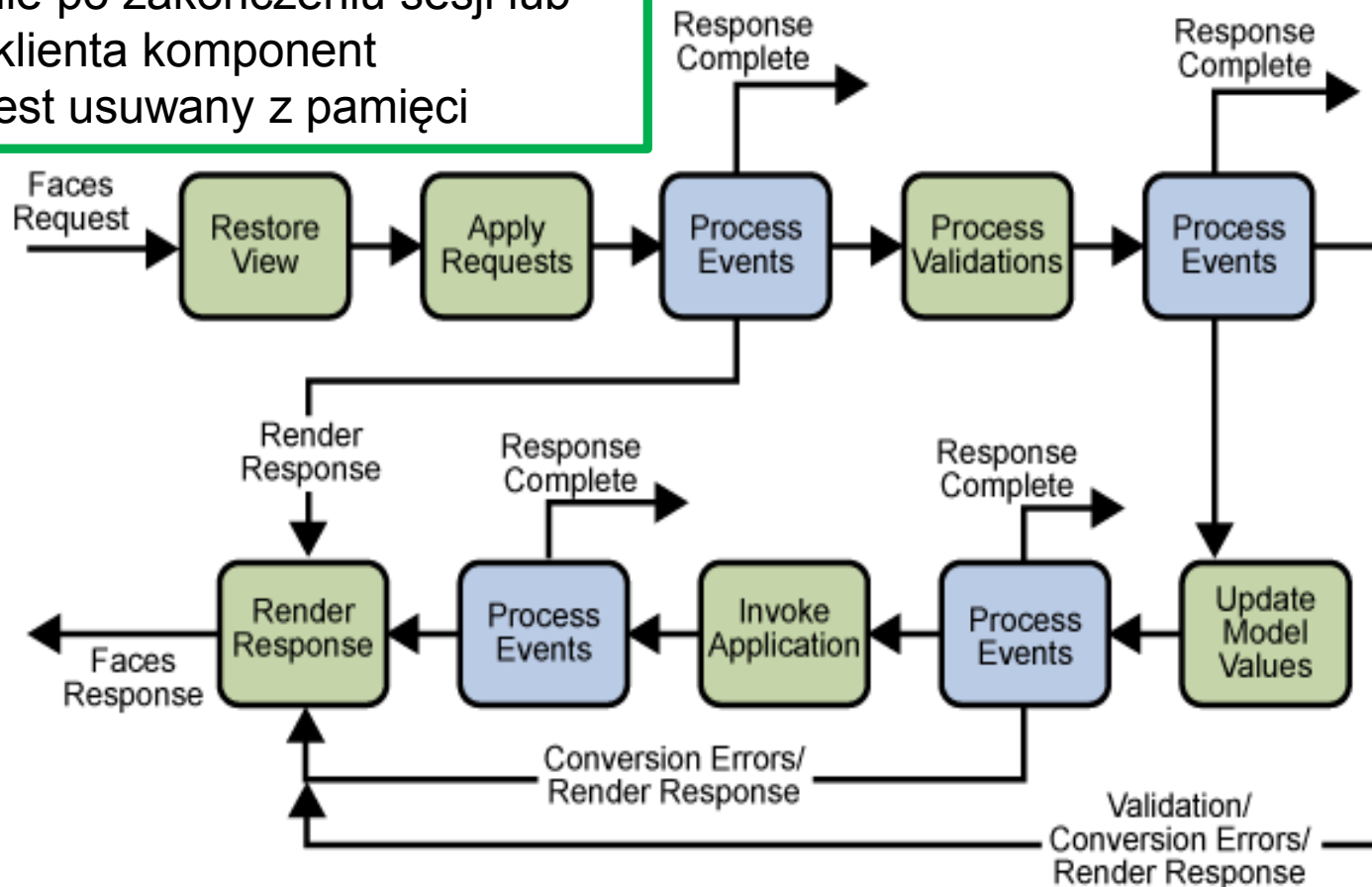
Powrot Odśwież stronę

Bottom

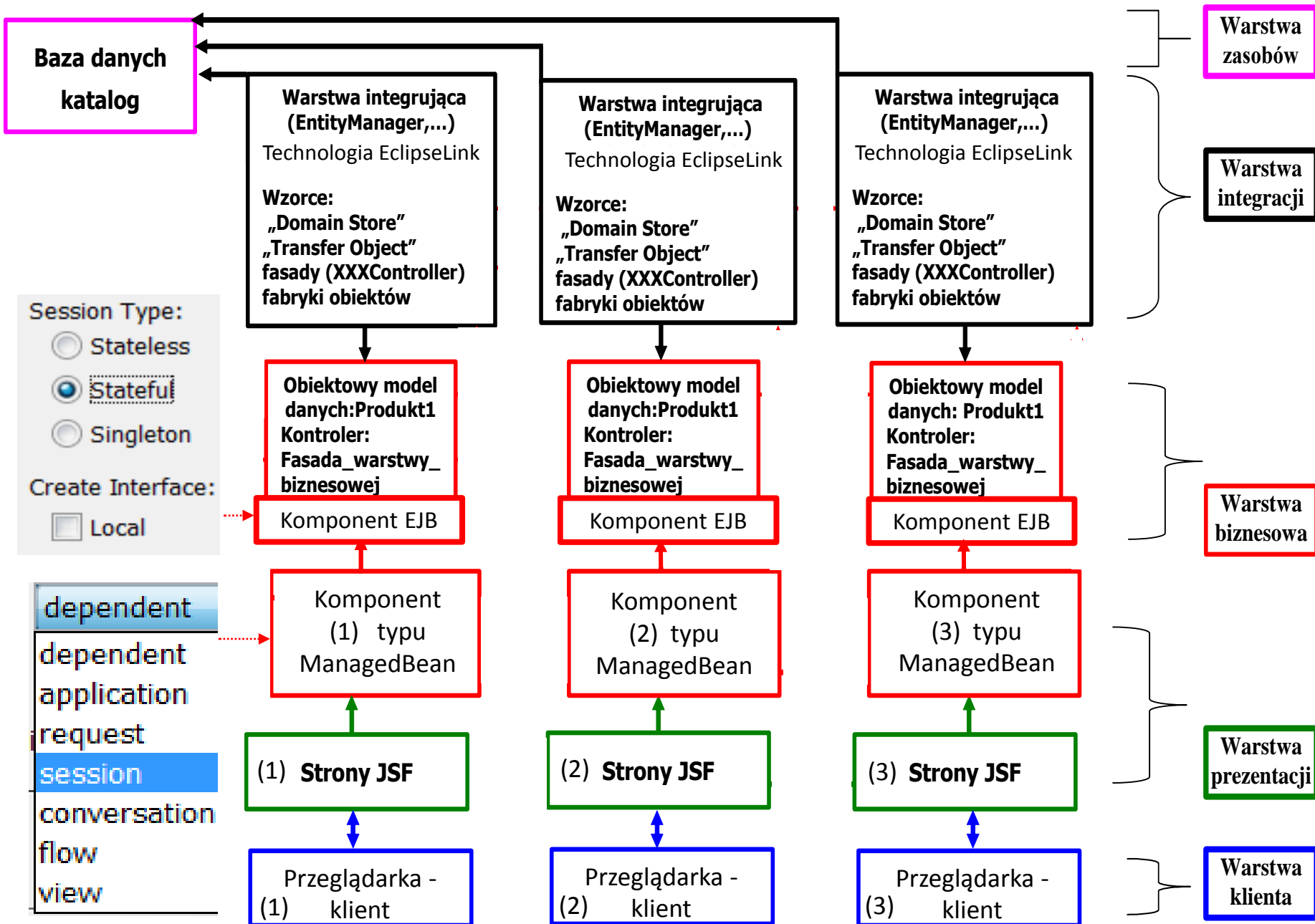
4. Wpływ komponentów typu **Managed Bean (SessionScoped)** i **Session Bean (Stateful)** na obsługę stronicowania i usuwania danych

Standard cyklu życia „Request-Response” dla JavaServer Faces

SessionScope: podczas uruchomienia instancji aplikacji klienta tworzony jest nowy komponent typu **Managed_produk**t – wszystkie jego dane są zaktualizowane podczas fazy Invoke Application. Po zakończeniu fazy Render Response komponent pozostaje w pamięci. Jeśli nie wystąpią błędy, jedynie po zakończeniu sesji lub zamknięciu aplikacji klienta komponent **Managed_produk**t jest usuwany z pamięci



Architektura aplikacji pięciowarstwowej – Java EE 7.0 JavaServer Faces



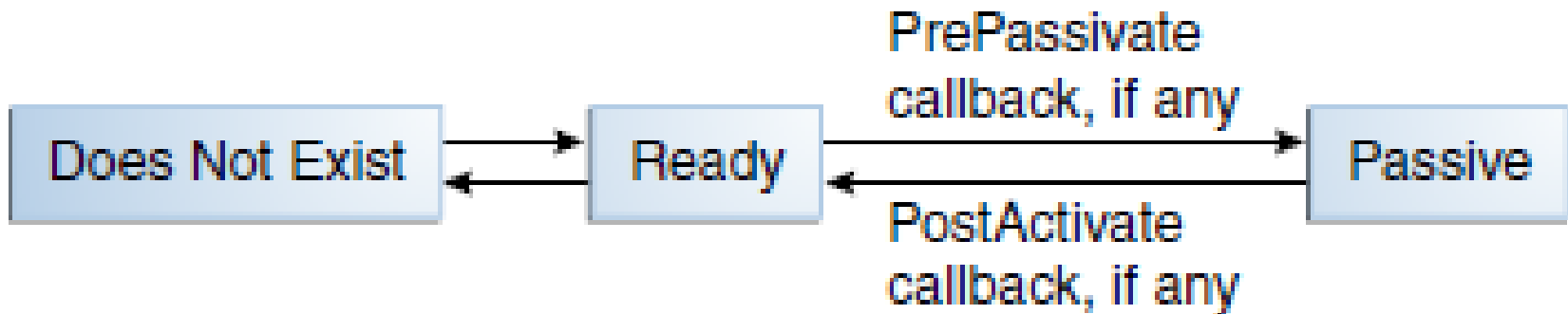
Prosta obsługa usuwania i stronicowania

1. Komponent typu Session Bean rodzaju Stateful przechowują dane jedynie jednej aplikacji klienckiej: internetowej lub dektopowej
2. Komponent typu Managed Bean przechowuje dane jedynie jednej aplikacji klienckiej: internetowej lub dektopowej
3. Takie rozwiązanie ogranicza jednak skalowalność aplikacji ze względu na duże zużycie zasobów przy dużej liczbie uruchomionych aplikacji klienckich: dektopowych i internetowych

Cykl życia stanowych (Stateful) komponentów typu Session Bean

Zadania kontenera EJB:

1. Tworzy komponent
2. Dependency injection, jeśli są
3. Metoda PostConstruct callback, jeśli jest
4. Metoda Init, lub ejbCreate<METHOD>, jeśli są



1. Usuwa komponent
2. Metoda PreDestroy callback, jeśli jest