

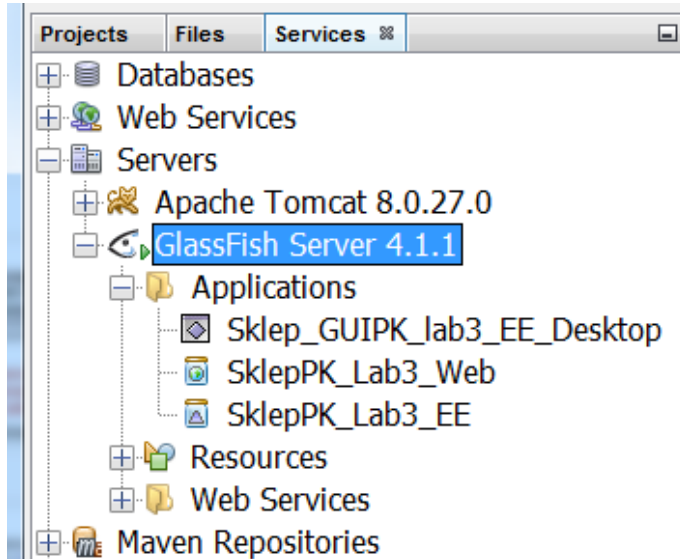
# **Kontynuacja programu z lab3 – zaawansowane renderowanie formularzy JSF na podstawie**

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

## **Programowanie komponentowe Lab4**

# 1. Przygotowanie projektów kopii projektów z lab3.

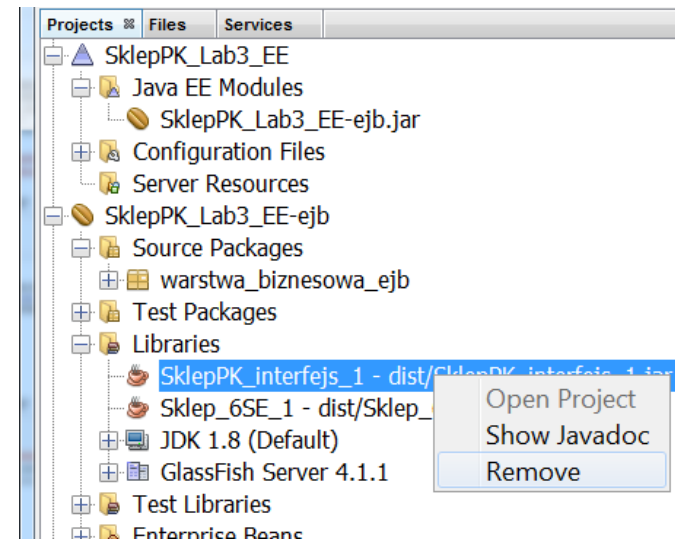
- 1) Należy wykonać operację **undeploy**, jeśli są uruchomione projekty, wykonane podczas laboratorium 3.



- 2) Należy zamknąć projekty (**Close**):

- 1) Sklep\_6SE\_1
- 2) SklepPK\_interfejs\_1
- 3) SklepPK\_Lab3\_EE-ejb
- 4) SklepPK\_Lab3\_Web
- 5) Sklep\_GUIPK\_lab3\_EE\_Desktop
- 6) SklepPK\_Lab3\_EE

3. Należy wykonać kopię katalogu głównego, w którym znajdują się projekty wykonane podczas laboratorium 3. Jeśli te programy są w różnych katalogach, należy je skopiować do jednego katalogu, a po uruchomieniu spróbować przywrócić powiązania pomiędzy projektami **wg p.1.3 z instrukcji do laboratorium 3**, usuwając dotychczasowe powiązania np. za pomocą pozycji **Remove** wybierając katalog **Libraries** projektu i zaznaczając usuwany projekt:



2. Wprowadzenie warunkowego renderowania strony [rezultat2.xhtml](#) w celu zapewnienia wieloużywalności tej strony – wyświetlanie danych wybranego produktu

## 2.1. Kod, który należy dodać do klasy **Fasada\_warstwy\_biznesowej**

Cel dodania kodu - eliminacja błędu wyświetlania strony **lista\_produkow.xhtml** w przypadku braku danych po wprowadzeniu stronicowania

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**.

```
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;  
}
```

W przypadku braku danych w kolekcji produktu metoda **getProdukty().get(0)** powoduje błąd, dlatego wprowadzono zabezpieczenie

2.2. 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>**

2.3. 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}"/>
```

2.4. 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.



## 2.5. Uzupełnienie zawartości pliku **Bundle.properties**

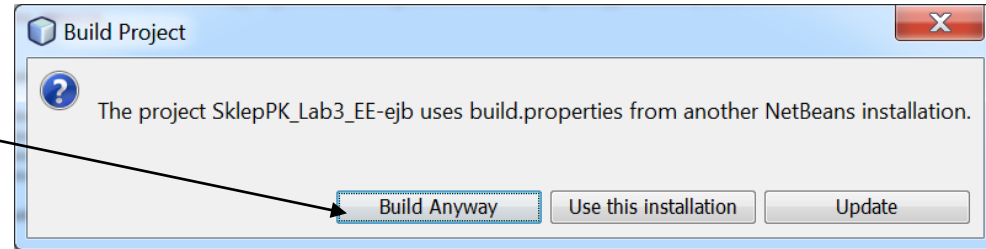
lista\_produkow.rezultat=Rezultat

rezultat2.akcja=Powrót

## 2.6. Uruchomienie projektu.

Należy w podanej kolejności wykonać operacje **Clean and Build** na projektach składowych (w celu łatwiejszej lokalizacji błędów):

- 1) Sklep\_6SE\_1
- 2) SklepPK\_interfejs\_1
- 3) SklepPK\_Lab3\_EE-ejb



- 4) SklepPK\_Lab3\_Web
- 5) Sklep\_GUIPK\_lab3\_EE\_Desktop
- 6) SklepPK\_Lab3\_EE

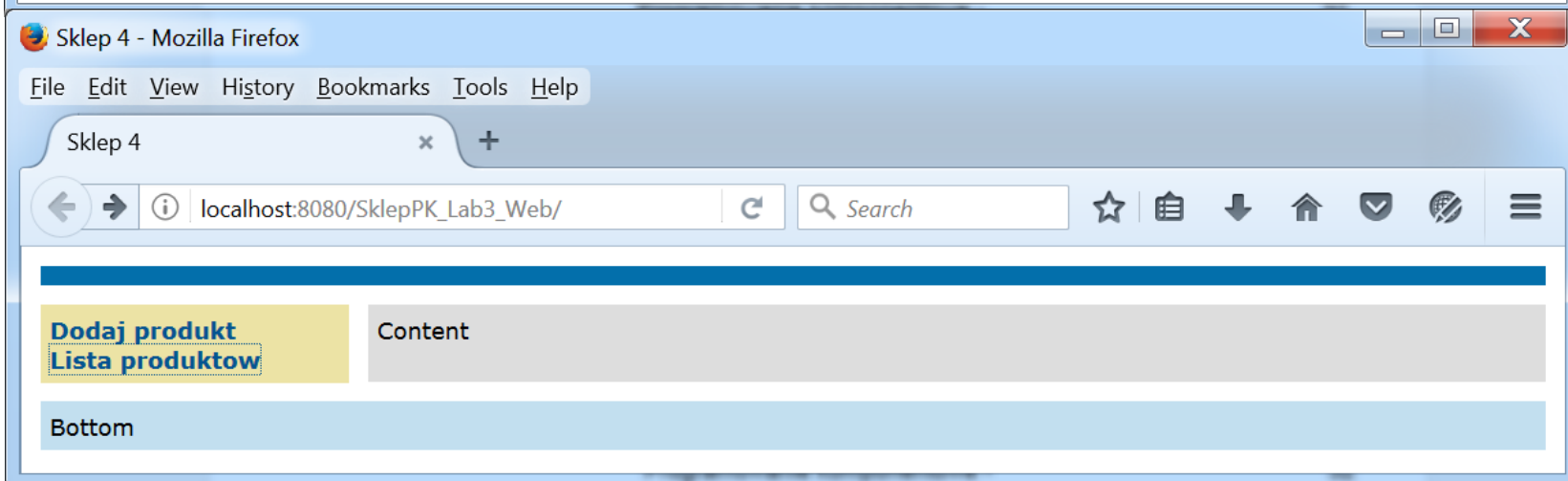
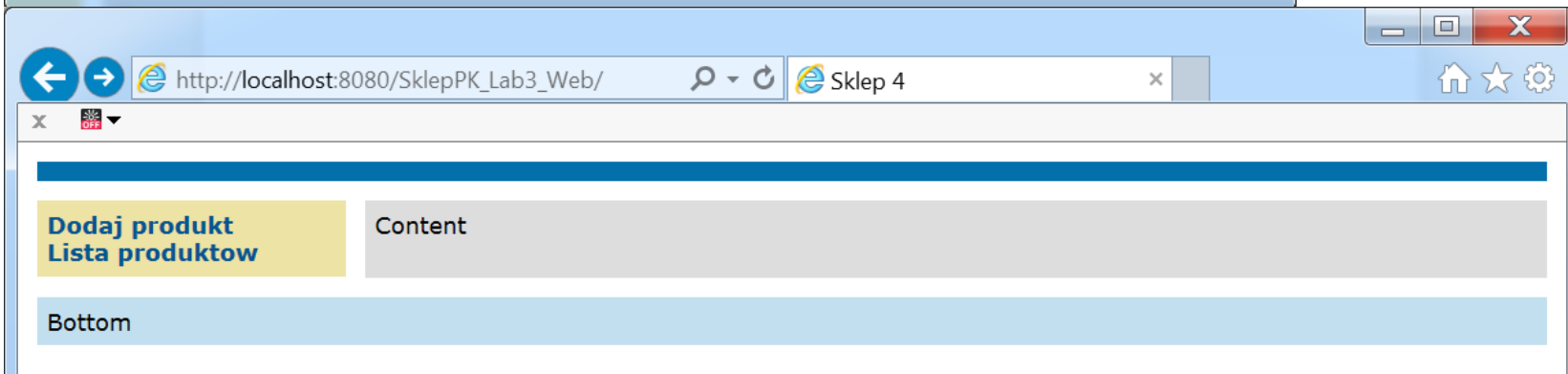
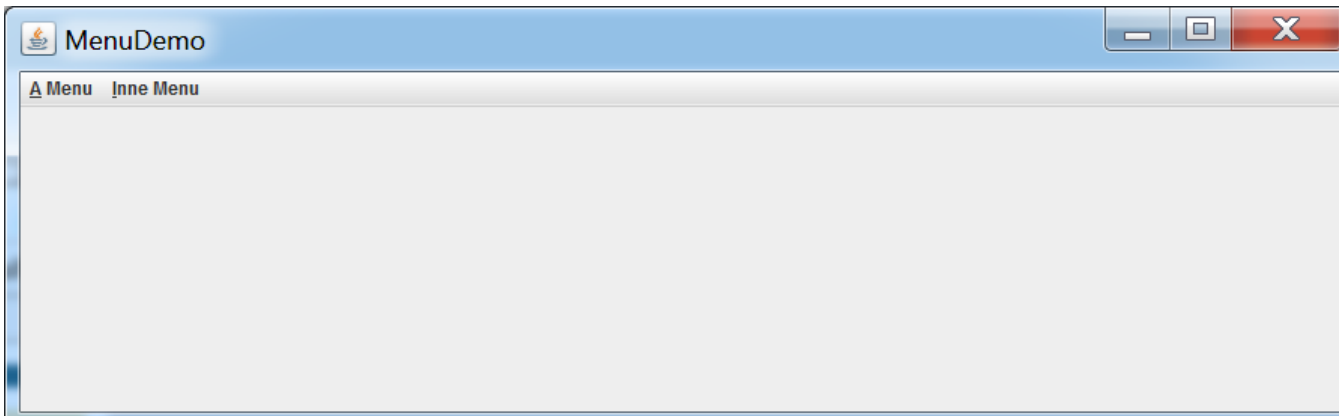
Następnie, należy wykonać operację **Deploy** na projekcie **SklepPK\_Lab3\_EE**.

Teraz można uruchomić dowolną liczbę aplikacji klienckich za pomocą operacji **Run**:

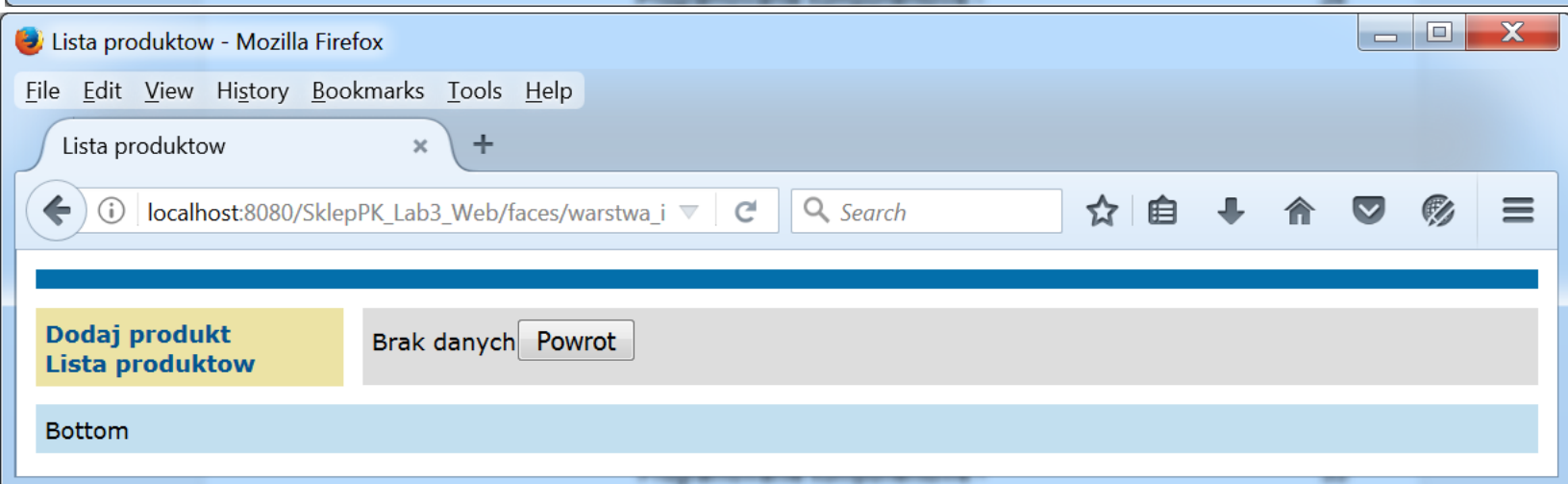
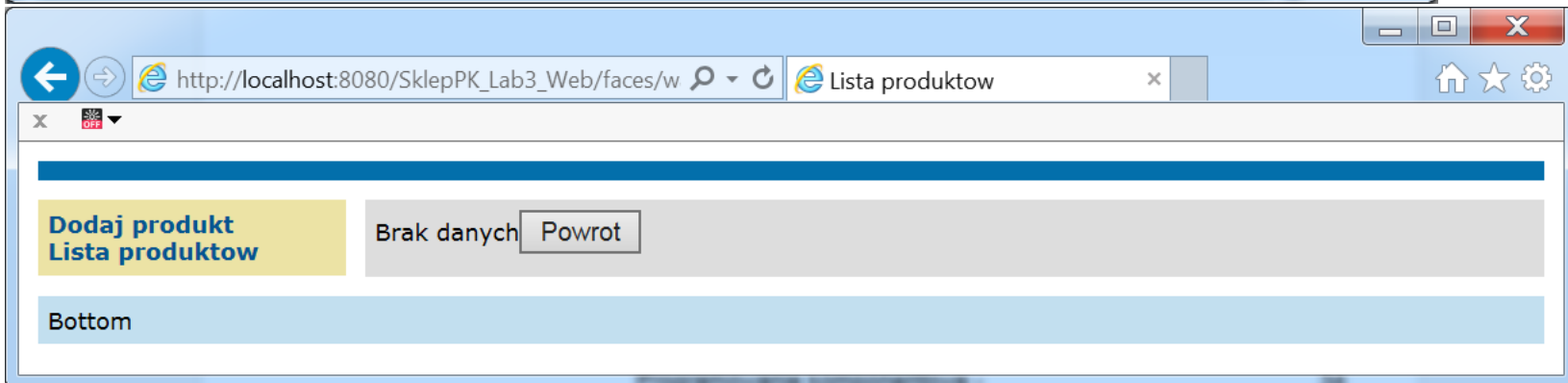
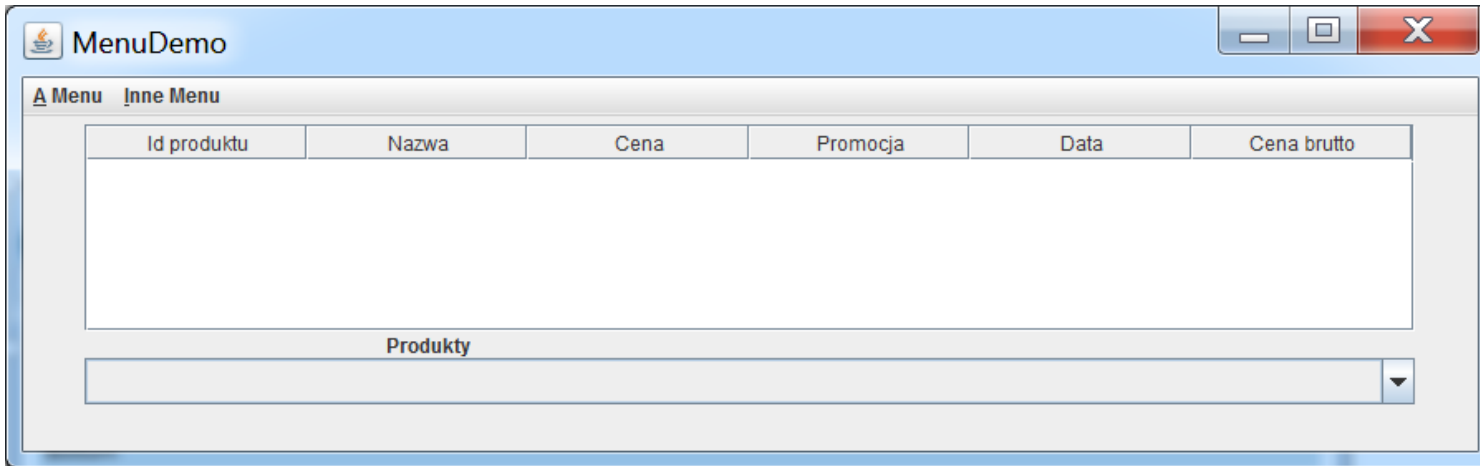
- 1) SklepPK\_Lab3\_Web (**pozostałe instancje w kolejnych instancjach przeglądark: [http://localhost:8080/ SklepPK\\_Lab3\\_Web/](http://localhost:8080/SklepPK_Lab3_Web/)**)
- 2) Sklep\_GUIPK\_lab3\_EE\_Desktop

W przykładzie uruchomiono jedną instancję aplikacji desktopowej i dwie instancje aplikacji internetowej.

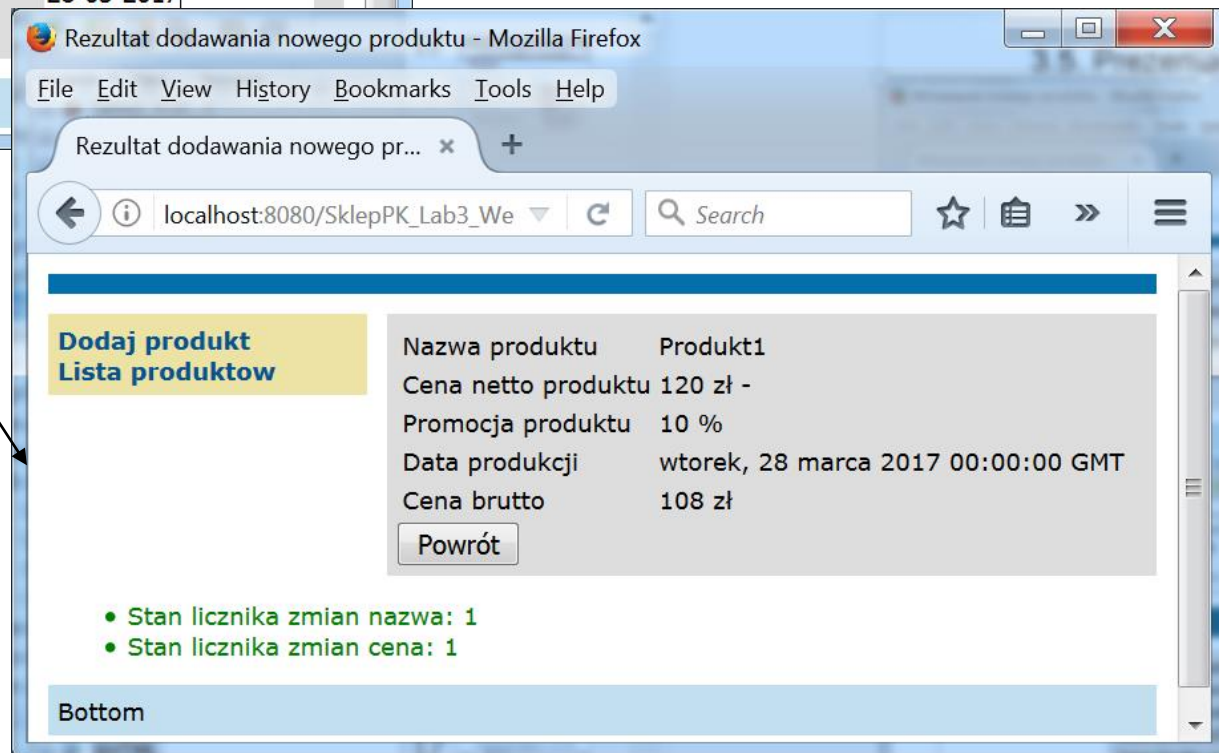
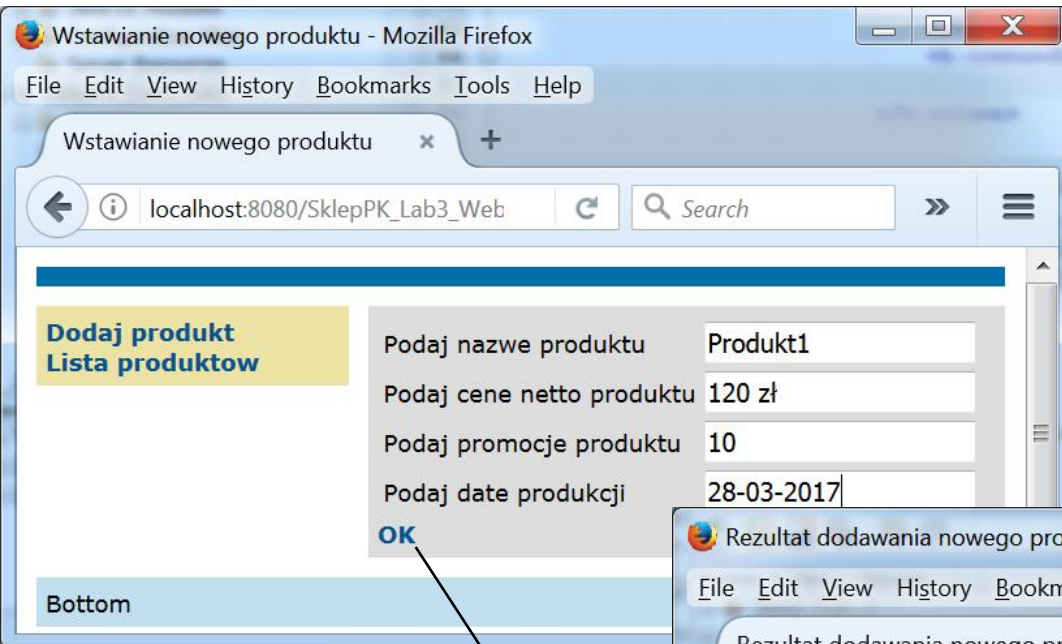
## 2.6. Prezentacja działania aplikacji w przypadku braku danych (1)



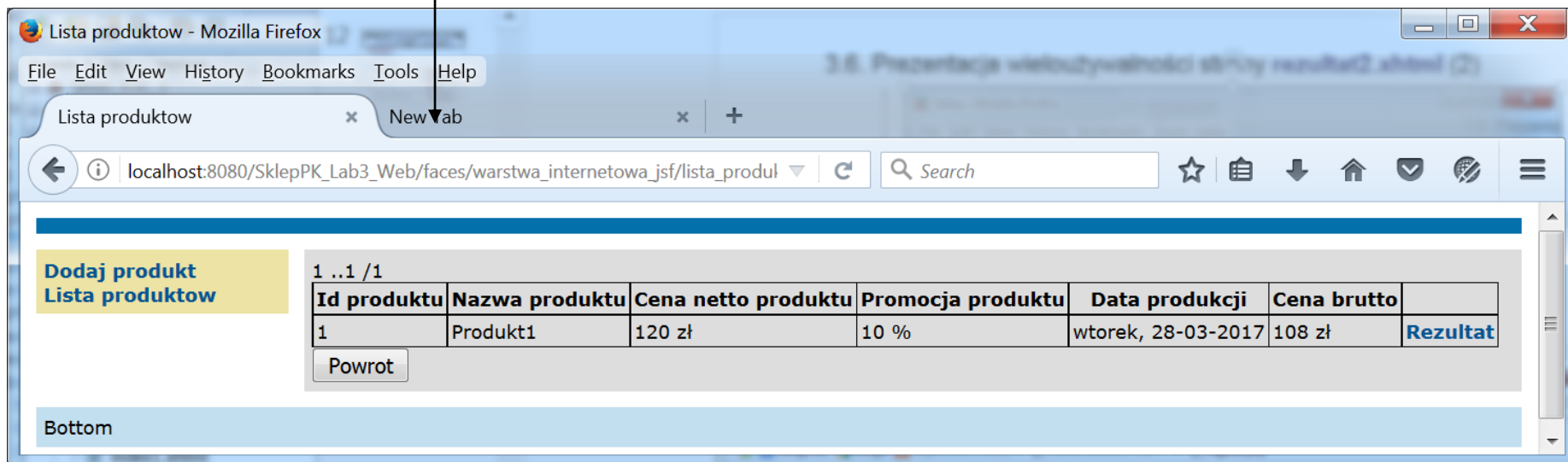
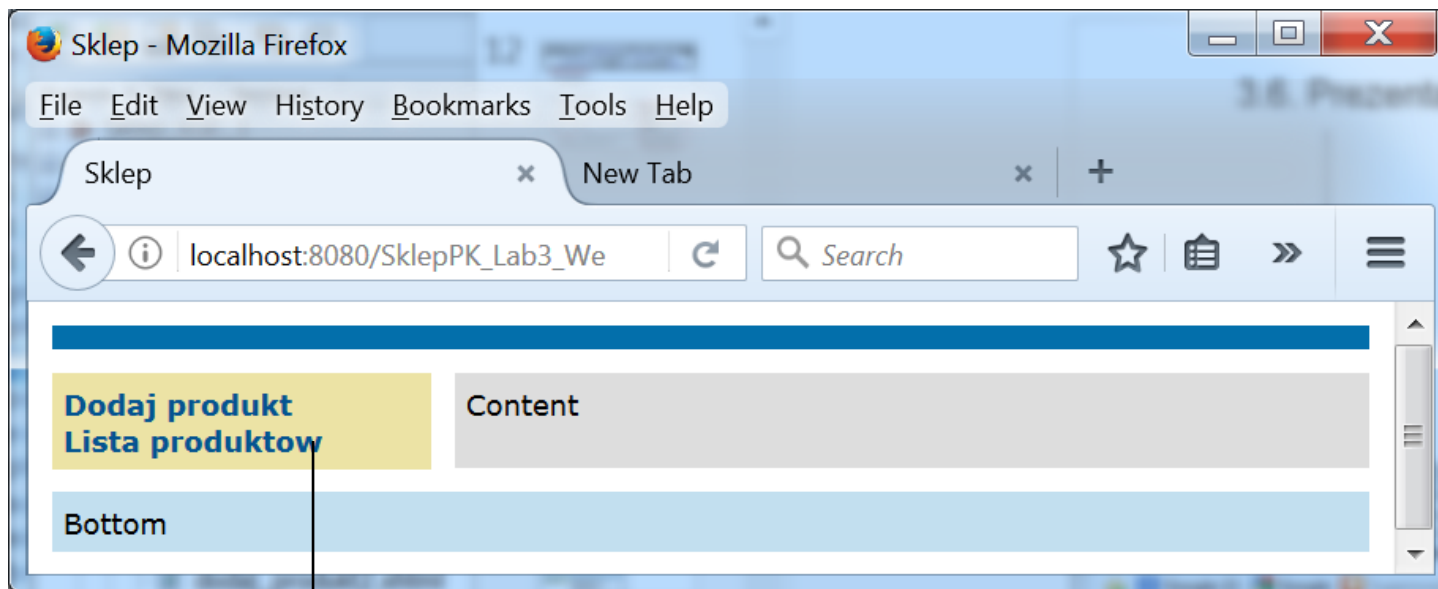
## 2.6. cd. Prezentacja działania aplikacji w przypadku braku danych (2)



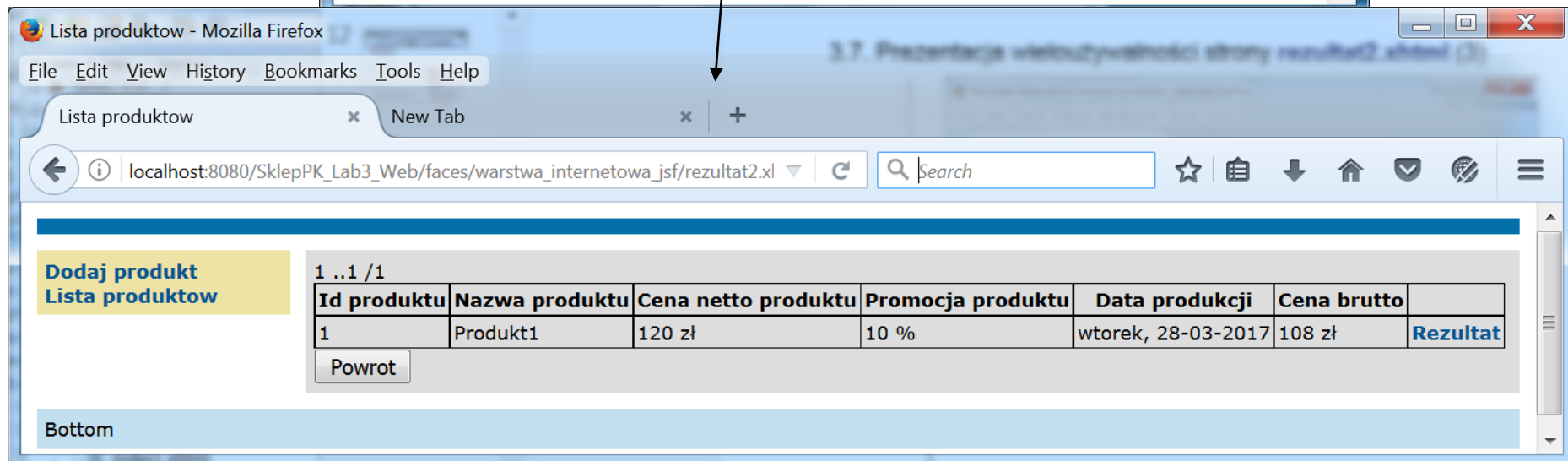
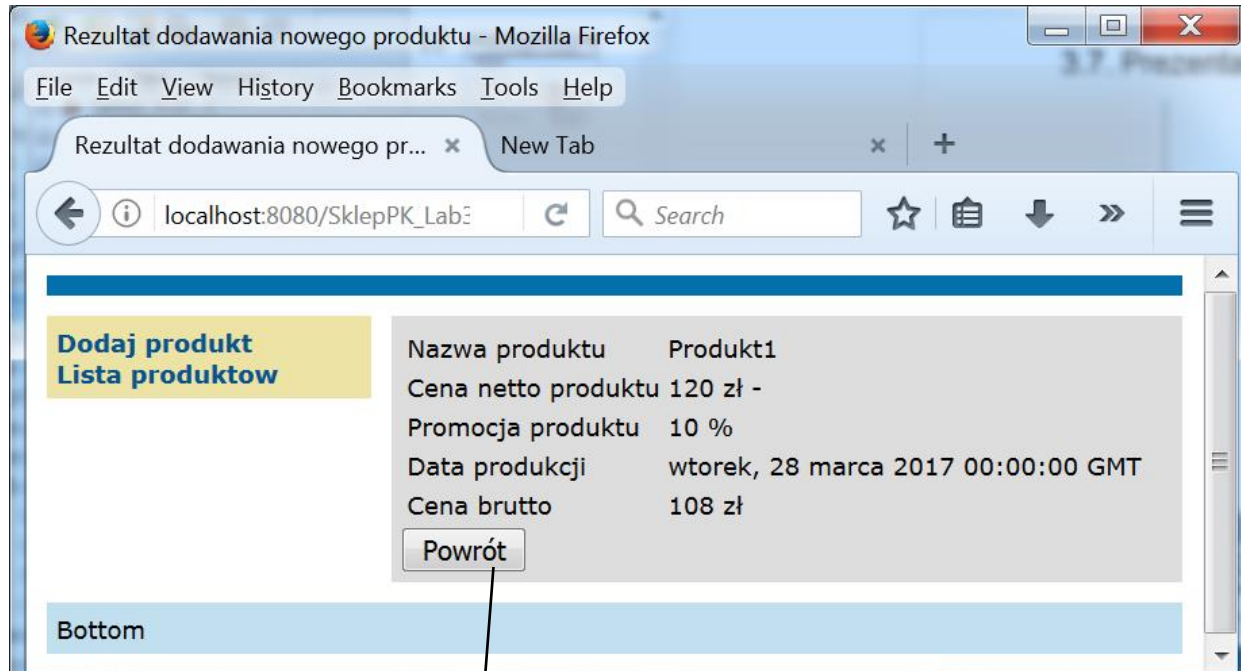
## 2.6. cd. Prezentacja wieloużywalności strony [rezulta2.xhtml](#) (1)- przejście na stronę [rezultat2.xhtml](#) po zakończeniu wprowadzania danych **OK** (3)



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](#)



3. Wprowadzenie warunkowego renderowania strony [dodaj\\_produk2.xhtml](#) w celu zapewnienia wieloużywalności tej strony – modyfikacja danych wybranego produktu



3.1. Dodanie przycisku wywołującego stronę **dodaj\_produk2.xhtml** w celu modyfikacji danych produktu z wybranego wiersza 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:column>
```

## 3.2. Dodanie nowych komunikatów do pliku [Bundle.properties](#)

Produkt\_zmieniony=Zmiana danych produktu

Blad\_modyfikacji=Nie dokonano zmiany danych produktu

lista\_produkow.edycja=Edycja

dodaj\_produk2.akcja=OK

3.3. 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);  
}
```

3.4. 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();  
}
```

### 3.5. 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; }

3.5. cd. 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,

3.5. cd. 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 (p. 3.7), 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_produk2ow";  
    } catch (Exception e) {  
        JsfUtil.addErrorMessage(e,  
        ResourceBundle.getBundle("/Bundle").getString("Blad_modyfikacji"));  
        return "lista_produk2ow"; }  
}
```

```
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();  
}
```



### 3.6. 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** (str. 23)

### 3.7. Modyfikacja metod w klasie **Fasada\_warstwy\_biznesowej**.

Metoda **edit** (str. 22) 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);  
}
```

## 3.7. cd

```
public void utworz_produkt(Produkt_dto produkt_dto) {  
    Produkt1 produkt = wykonaj_produkt(produkt_dto);  
    dodaj_produkt(produkt);  
}
```

```
Produkt1 wykonaj_produkt(Produkt_dto produkt_dto) {  
    Produkt1 produkt = new Produkt1();  
    max_klucz();  
    produkt.setId(new Long(klucz));  
    produkt.setNazwa(produkt_dto.getNazwa());  
    produkt.setCena(produkt_dto.getCena());  
    produkt.setPromocja(produkt_dto.getPromocja());  
    produkt.setData_produkcji(produkt_dto.getData_produkcji());  
    return produkt;  
}
```

```
void max_klucz() {  
    long max = 0;  
    for (Produkt1 p : produkty)  
        if (p.getId() > max) max = p.getId();  
    klucz = max + 1; }  
}
```

Refaktoryzacja metody **utworz\_produkt** przez dodanie metody **wykonaj\_produkt**,

Metoda **max\_klucz** wyznacza wartość kolejnego największego klucza zapewniając jego unikatowość,

```
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;  
}
```

### 3.8. 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);
}
```

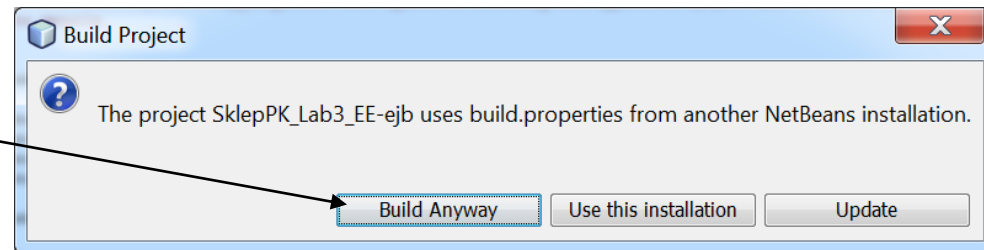
### 3.8. cd. Implementacja nowej metody **edit** w komponencie EJB – **Fasada\_warstwy\_biznesowej\_ejb**

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

## 3.9. Uruchomienie projektu.

Należy w podanej kolejności wykonać operacje **Clean and Build** na projektach składowych (w celu łatwiejszej lokalizacji błędów):

- 1) Sklep\_6SE\_1
- 2) SklepPK\_interfejs\_1
- 3) SklepPK\_Lab3\_EE-ejb



- 4) SklepPK\_Lab3\_Web
- 5) Sklep\_GUIPK\_lab3\_EE\_Desktop
- 6) SklepPK\_Lab3\_EE

Następnie, należy wykonać operację **Deploy** na projekcie **SklepPK\_Lab3\_EE**.

Teraz można uruchomić dowolną liczbę aplikacji klienckich za pomocą operacji **Run**:

- 1) SklepPK\_Lab3\_Web (**pozostałe instancje w kolejnych instancjach przeglądark: [http://localhost:8080/ SklepPK\\_Lab3\\_Web/](http://localhost:8080/SklepPK_Lab3_Web/)**)
- 2) Sklep\_GUIPK\_lab3\_EE\_Desktop

W przykładzie uruchomiono jedną instancję aplikacji desktopowej i dwie instancje aplikacji internetowej.

### 3.9. 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	<a href="#">Rezultat Edycja Usun</a>
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	<a href="#">Rezultat Edycja Usun</a>
3	Produkt4	220 zł	20 %	piątek, 28-04-2017	176 zł	<a href="#">Rezultat Edycja Usun</a>

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



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

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktów

localhost:8080/SklepPK\_Lab3\_Web/faces/warstwa\_internetowa\_jsf/dodaj\_produk2.xhtml

**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	<a href="#">Rezultat Edycja Usun</a>
2	Produkt2	123 zł	12 %	piątek, 28-04-2017	108,24 zł	<a href="#">Rezultat Edycja Usun</a>
3	Produkt4	220 zł	20 %	piątek, 28-04-2017	176 zł	<a href="#">Rezultat Edycja Usun</a>

Powrot

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

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

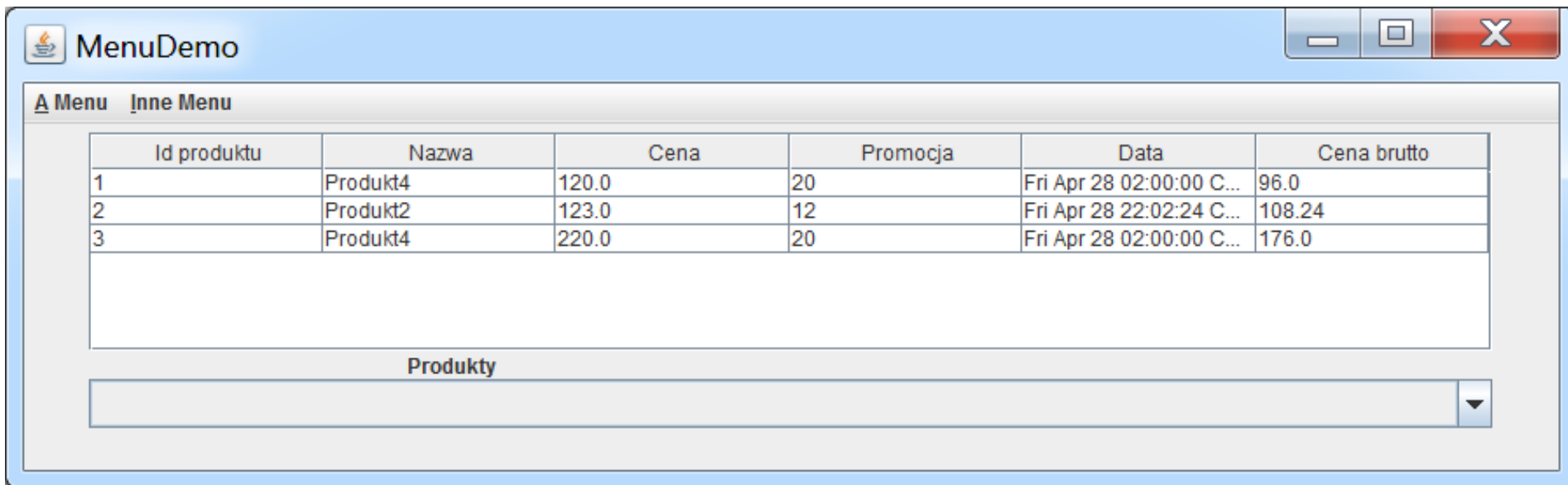
1 ..3 /3

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

Powrot

Bottom

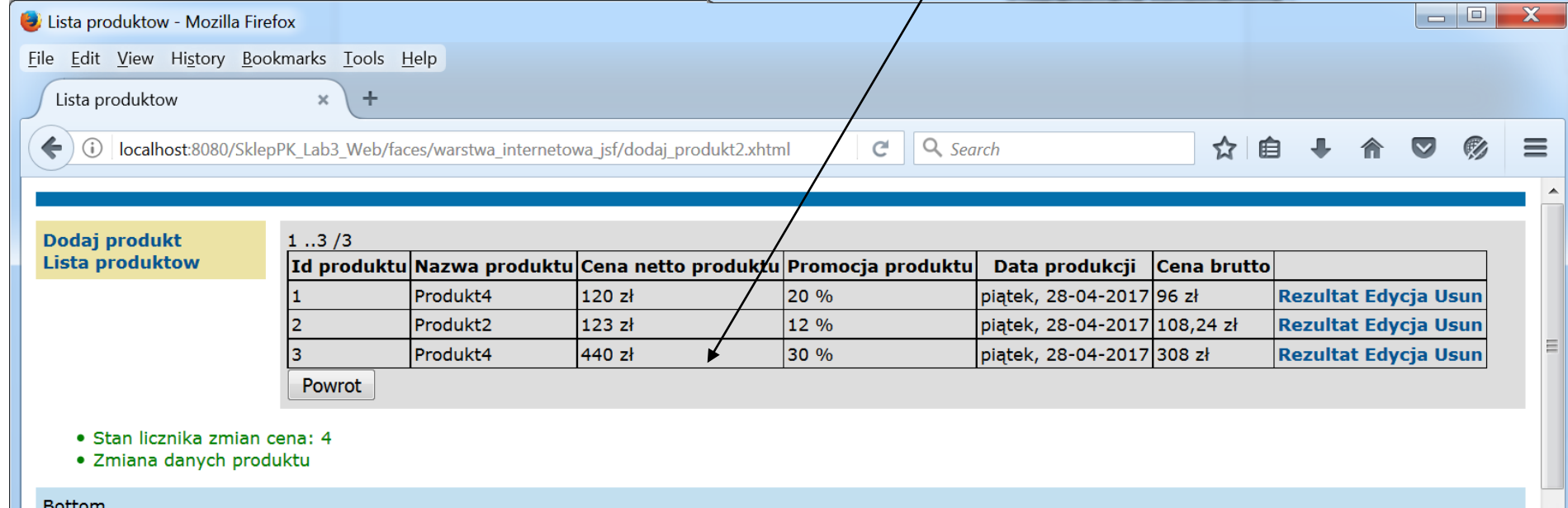
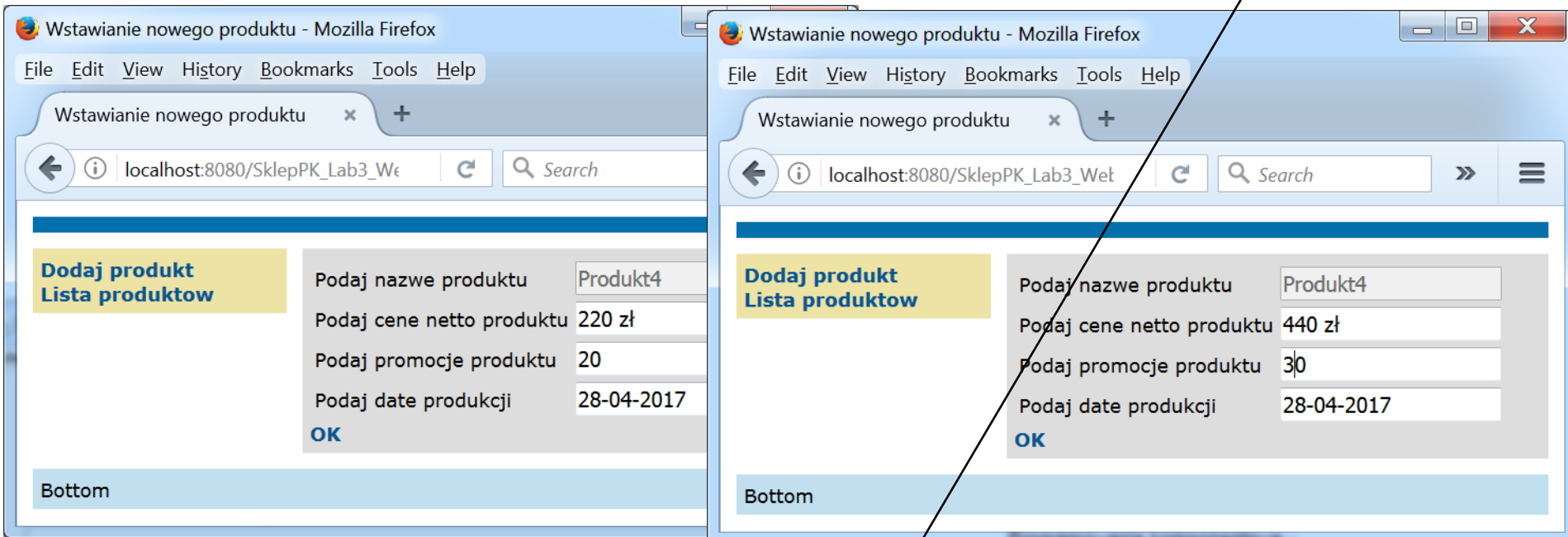
### 3.9. cd. Prezentacja procesu modyfikacji danych (3)



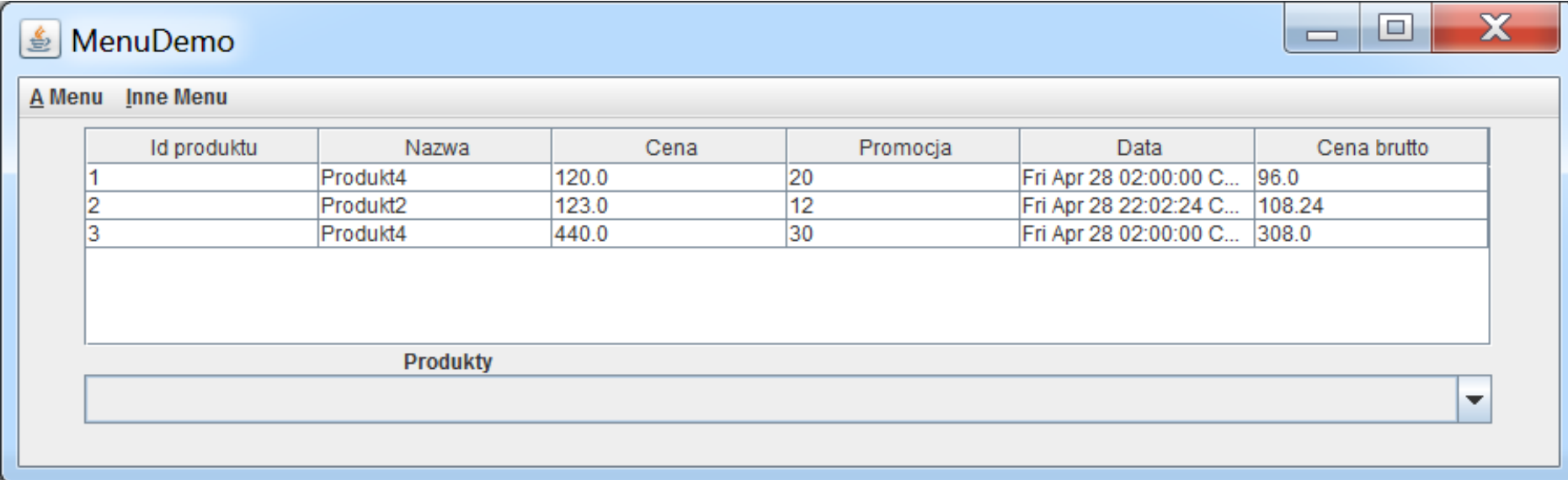
Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt4	120.0	20	Fri Apr 28 02:00:00 C...	96.0
2	Produkt2	123.0	12	Fri Apr 28 22:02:24 C...	108.24
3	Produkt4	220.0	20	Fri Apr 28 02:00:00 C...	176.0

Produkty

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



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



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

4. Dodanie na stronie  
**lista\_produkow.xhtml** przycisku do  
usuwania danych

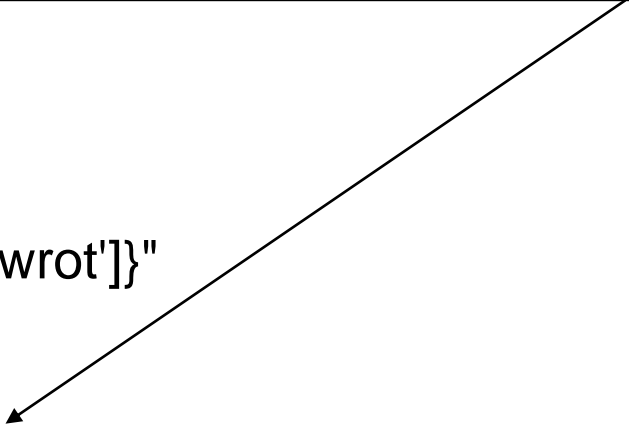
4.1. Dodanie przycisku do usuwania produktu wybranego w wierszu tabeli typu **dataTable** na stronie **lista\_produkow.xhtml**

```
<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>
```

4.2. 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}" />
```



### 4.3. Uzupełnienie zawartości pliku **Bundle.properties**

Usunieto\_produk=Produkt został usuniety

Blad\_usuwania=Produkt nie został usuniety

lista\_produk.usun=Usun

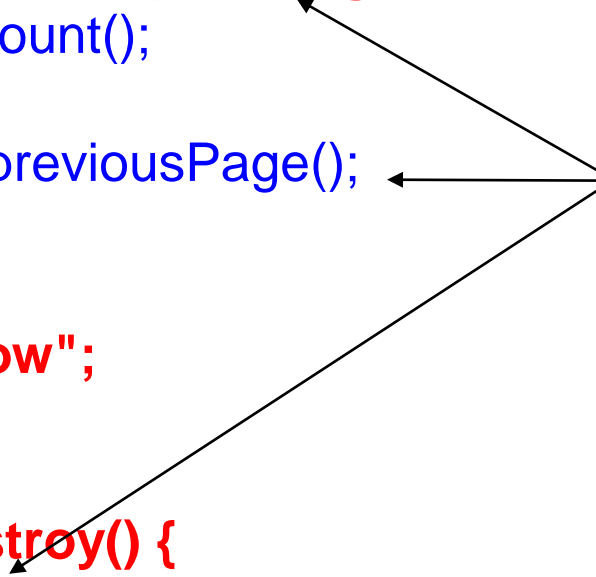
lista\_produk.refresh=Odśwież stronę



## 4.4. 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_produkow";  
}
```

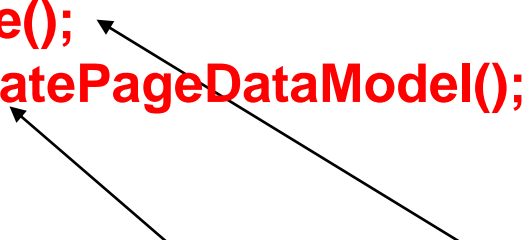
Przygotowanie  
aktualizacji strony po  
usunięciu wybranego  
wiersza tabeli



```
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"));  
    }  
}
```

4.5. 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
```



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

4.6. 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**

## 4.7. Dodanie metody **remove** do klasy **Fasada\_warstwy\_biznesowej**

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

## 4.8. 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);
}
```

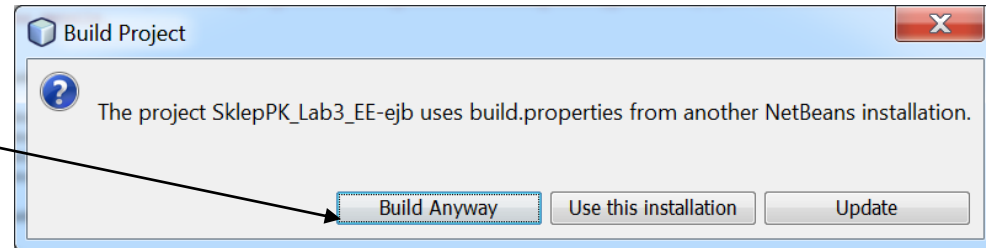
## 4.8. cd. Implementacja nowej metody **remove** w komponencie EJB – **Fasada\_warstwy\_biznesowej\_ejb**

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

## 4.9. Uruchomienie projektu.

Należy w podanej kolejności wykonać operacje **Clean and Build** na projektach składowych (w celu łatwiejszej lokalizacji błędów):

- 1) Sklep\_6SE\_1
- 2) SklepPK\_interfejs\_1
- 3) SklepPK\_Lab3\_EE-ejb



- 4) SklepPK\_Lab3\_Web
- 5) Sklep\_GUIPK\_lab3\_EE\_Desktop
- 6) SklepPK\_Lab3\_EE

Następnie, należy wykonać operację **Deploy** na projekcie **SklepPK\_Lab3\_EE**.

Teraz można uruchomić dowolną liczbę aplikacji klienckich za pomocą operacji **Run**:

- 1) SklepPK\_Lab3\_Web (**pozostałe instancje w kolejnych instancjach przeglądark: [http://localhost:8080/ SklepPK\\_Lab3\\_Web/](http://localhost:8080/SklepPK_Lab3_Web/)**)
- 2) Sklep\_GUIPK\_lab3\_EE\_Desktop

W przykładzie uruchomiono jedną instancję aplikacji desktopowej i dwie instancje aplikacji internetowej.

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

The image shows two overlapping windows. The top window is a Java Swing application titled "MenuDemo". It has a menu bar with "Menu" and "Inne Menu". Below the menu bar is a table with 6 columns: "Id produktu", "Nazwa", "Cena", "Promocja", "Data", and "Cena brutto". The table contains 5 rows of product data. Below the table is a label "Produkty" and a scrollable area.

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

The bottom window is a Mozilla Firefox browser titled "Lista produktow". The address bar shows "localhost:8080/SklepPK\_Lab3\_Web/faces/warstwa\_internetowa\_jsf/lista\_produkow.xhtn". The page content includes a sidebar with "Dodaj produkt" and "Lista produktow". The main content area shows a table with 7 columns: "Id produktu", "Nazwa produktu", "Cena netto produktu", "Promocja produktu", "Data produkcji", "Cena brutto", and "Rezultat Edycja Usun". The table shows products 4 and 5. Below the table are buttons for "Powrot" and "Odśwież stronę".

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	Rezultat Edycja Usun
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

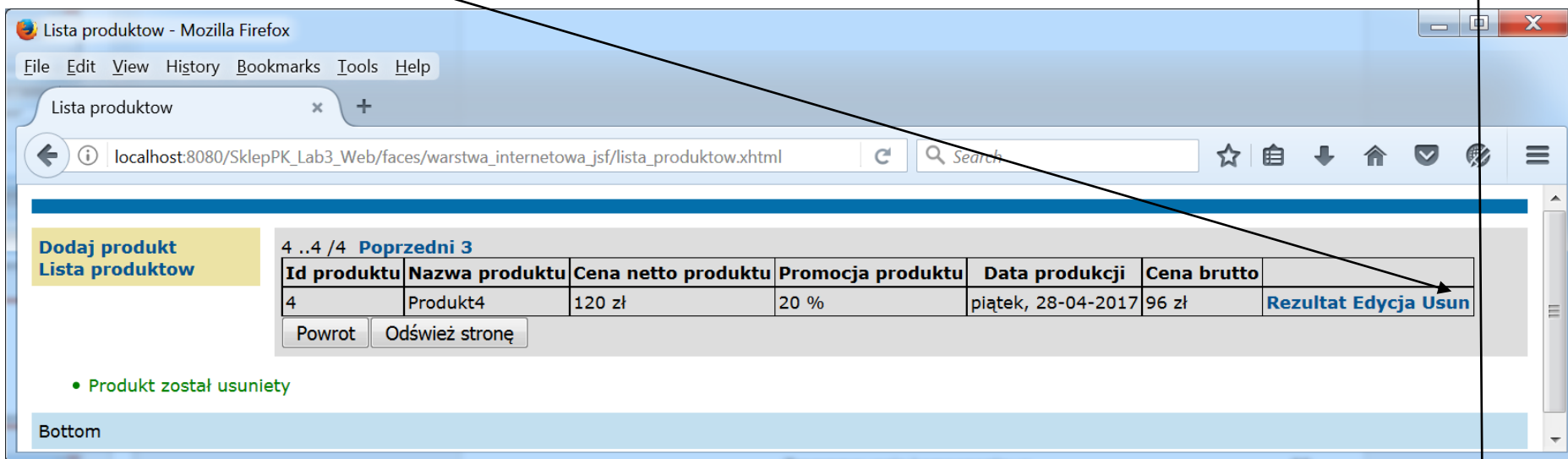


4.10. cd. Widok danych na stronie **lista\_produkow** w drugiej instancji klienta internetowego (**klient2**) – domyślny i po kliknięciu na przycisk **Następny 3** (2)

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
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

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

## 4.10. 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

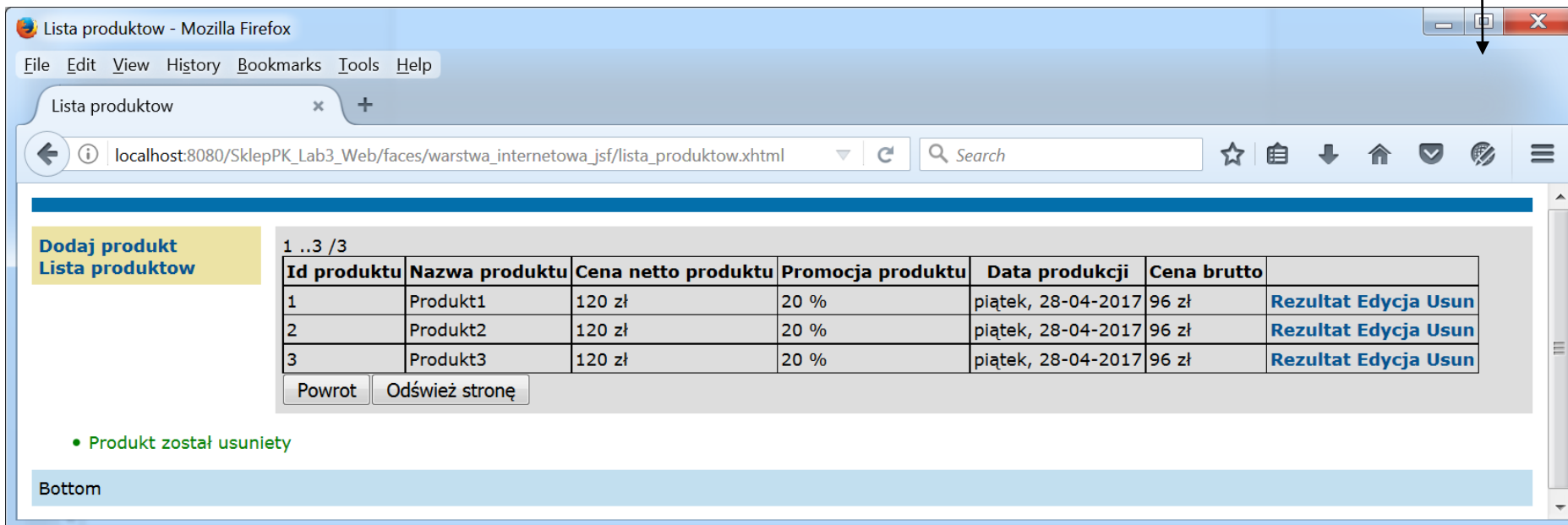
**Dodaj produkt**  
**Lista produktow**

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	<a href="#">Rezultat Edycja Usun</a>

• 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	<a href="#">Rezultat Edycja Usun</a>
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>

• Produkt został usuniety

Bottom

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

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

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

## 4.10. cd. Kolejne usuwanie wiersza tabeli na stronie **lista\_produkтов klienta1**. Strona **lista\_produkтов klienta2** wymaga aktualizacji. (5)

Lista produktow - Mozilla Firefox

Lista produktow

localhost:8080/SklepPK\_Lab3\_Web/faces/warstwa\_internetowa\_jsf/lista\_produkтов.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	<a href="#">Rezultat Edycja Usun</a>
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>

Powrot Odśwież stronę

• Produkt został usuniety

Bottom

http://localhost:8080/SklepPK\_Lab3\_Web/faces/warstwa\_internetowa\_jsf/li: Lista produktow

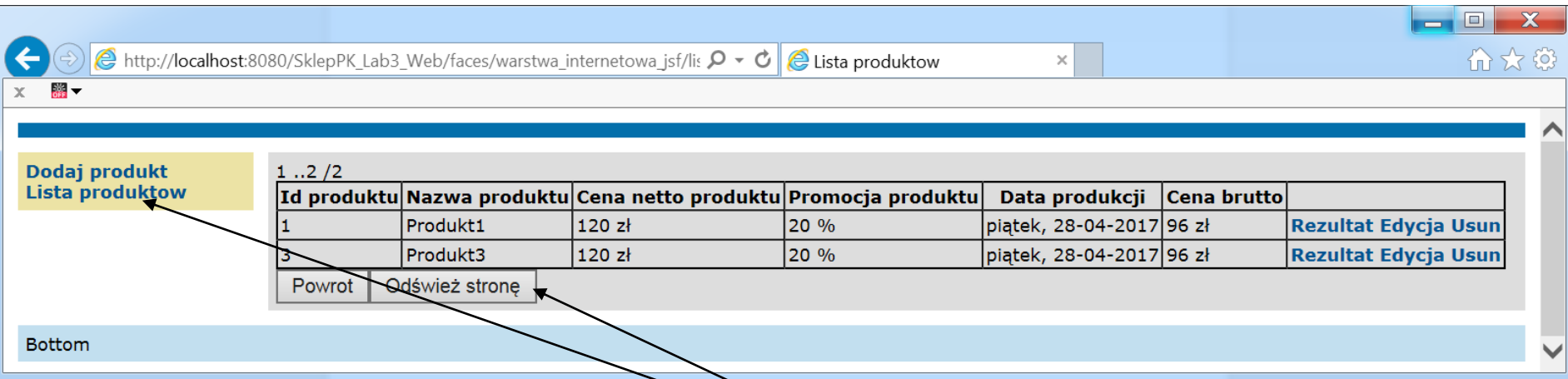
**Dodaj produkt**  
**Lista produktow**

1 .. 3 / 3

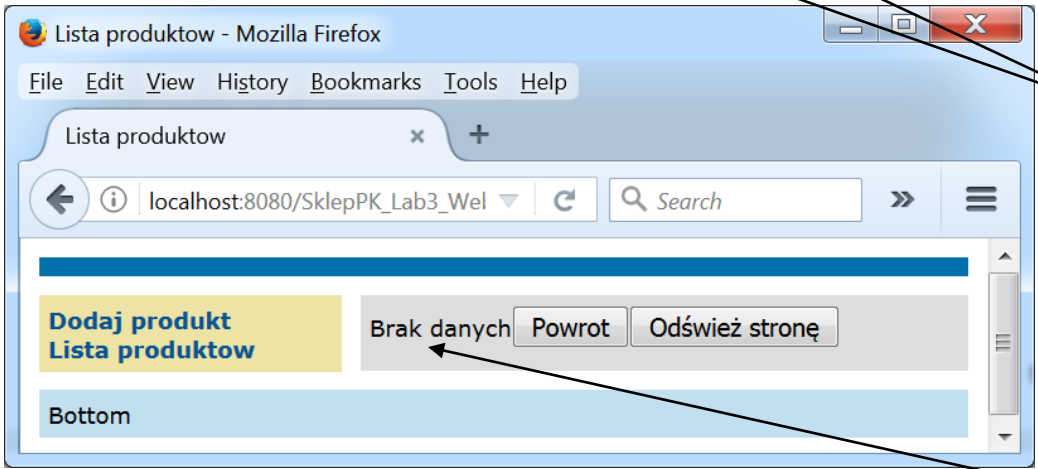
Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto	
1	Produkt1	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>
2	Produkt2	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>
3	Produkt3	120 zł	20 %	piątek, 28-04-2017	96 zł	<a href="#">Rezultat Edycja Usun</a>

Powrot Odśwież stronę

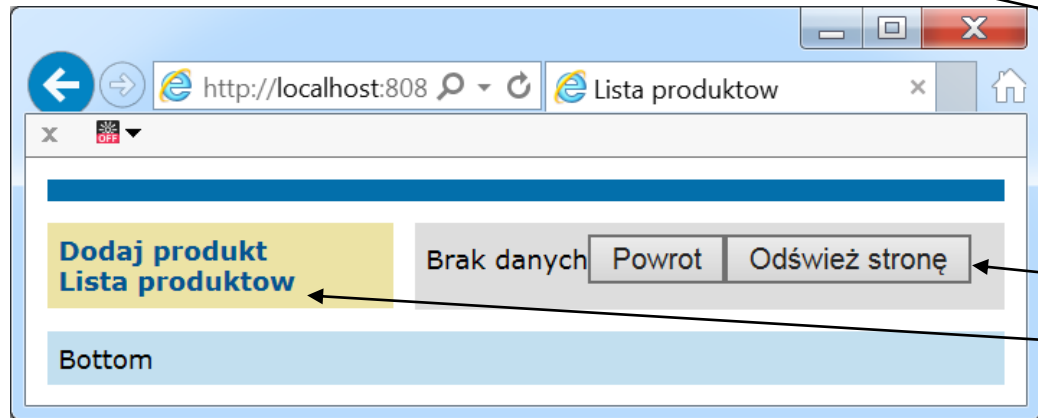
Bottom



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



4.10. 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)



5. Należy zmodyfikować stronę lista\_produkow.xhtml przez zastosowanie technologii Ajax