

Podstawowe informacje o technologii Java Persistence API - przykład

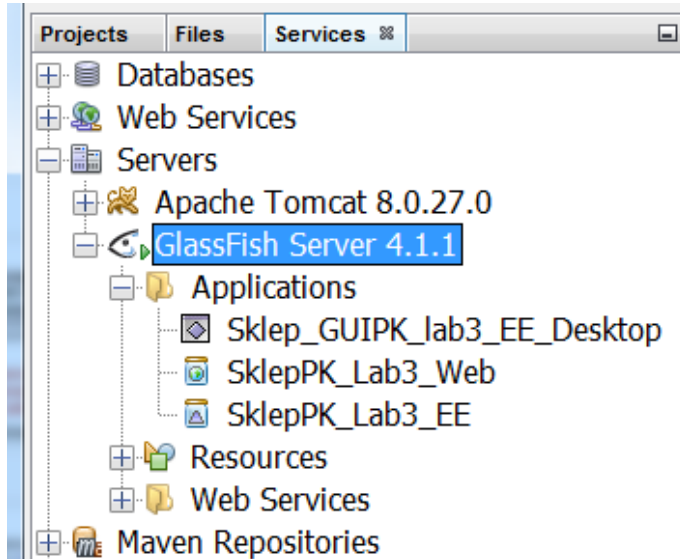
na podstawie

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

**Programowanie
komponentowe
lab5**

1 (1). Przygotowanie projektów kopii projektów z lab4.

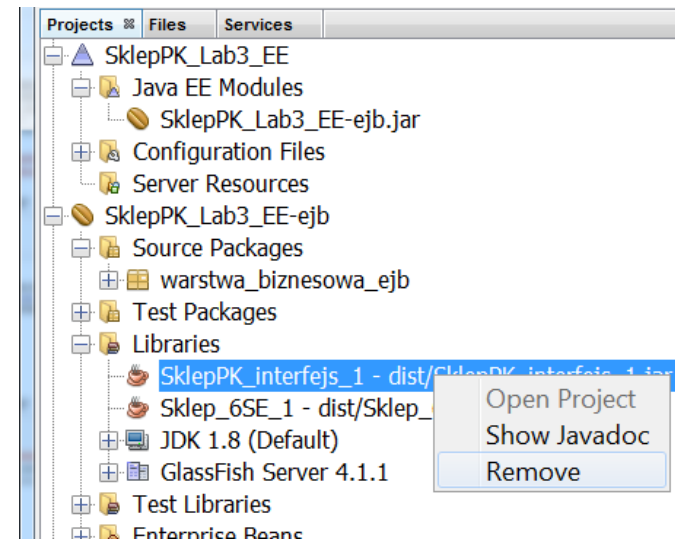
- 1) Należy wykonać operację **undeploy**, jeśli są uruchomione projekty, wykonane podczas laboratorium 4.



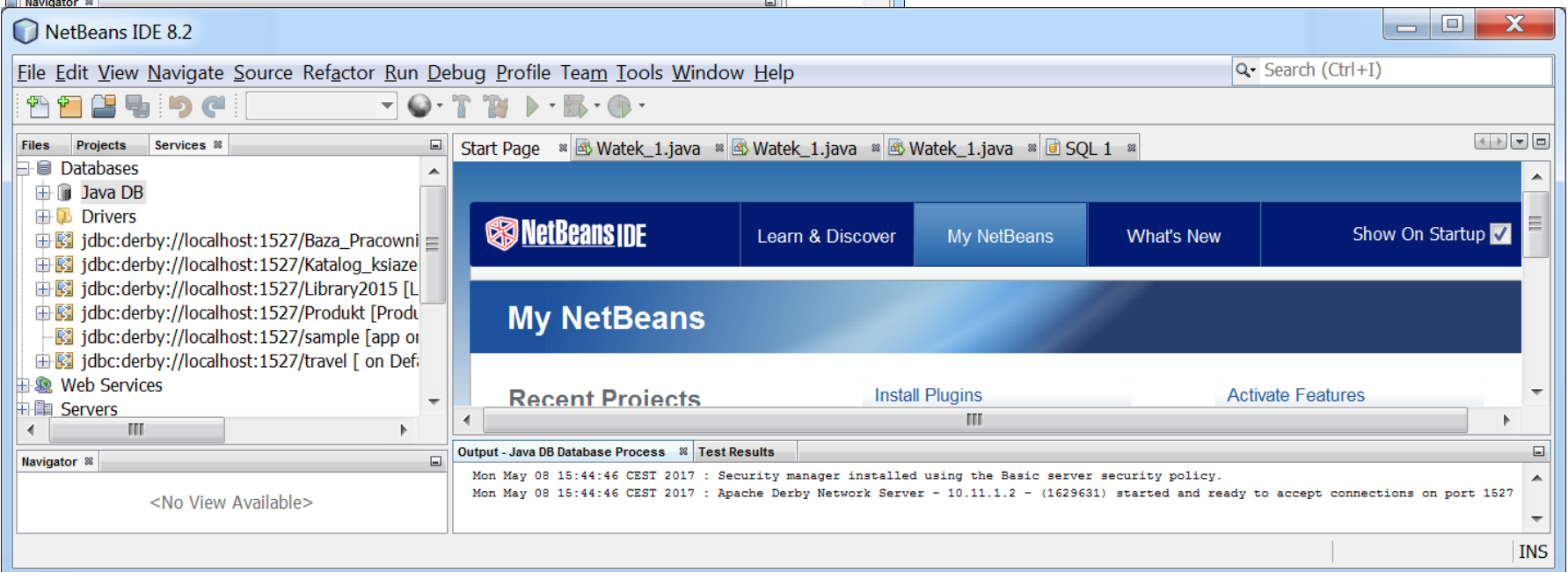
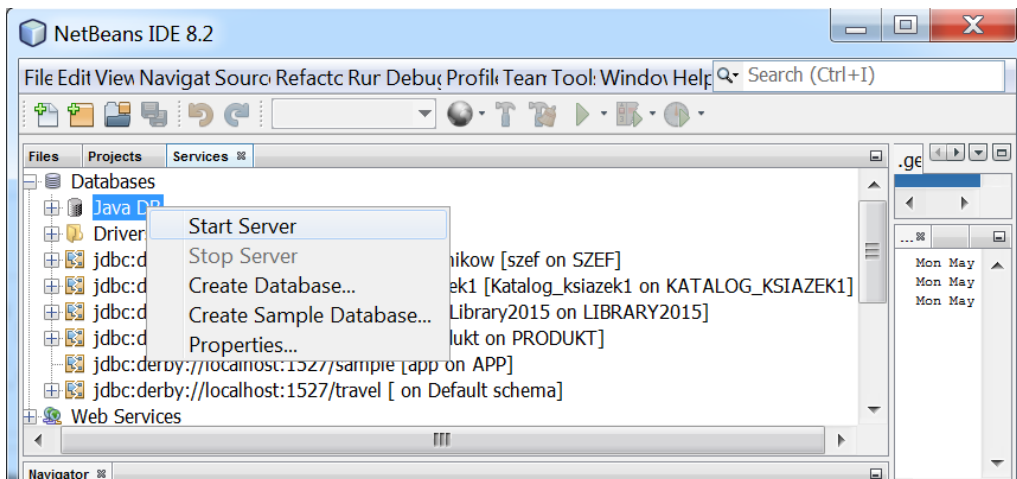
- 2) Należy zamknąć projekty (**Close**):

- 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

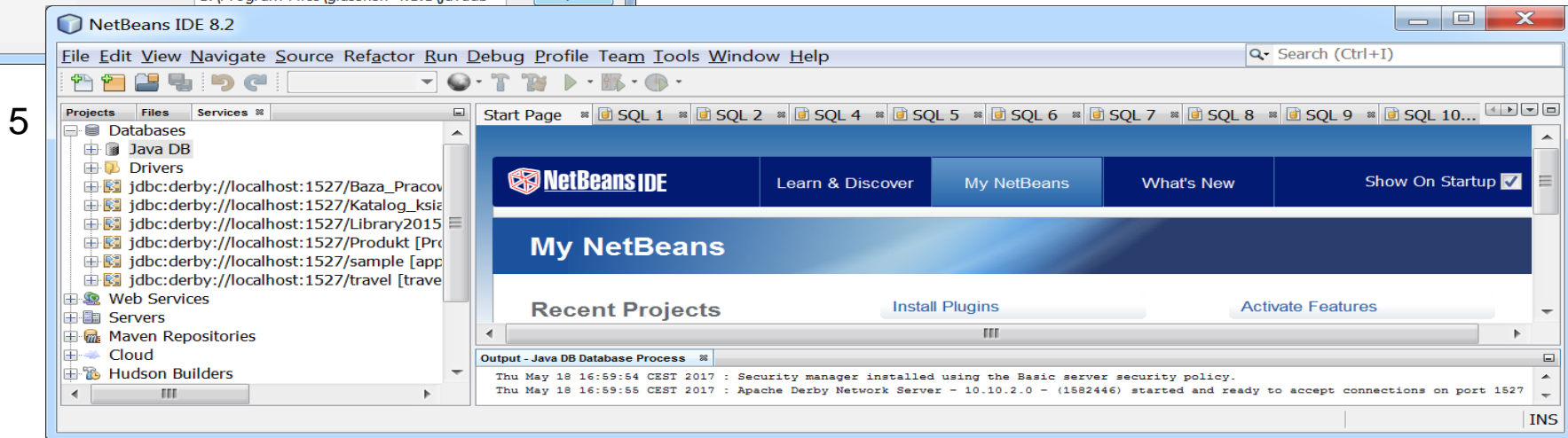
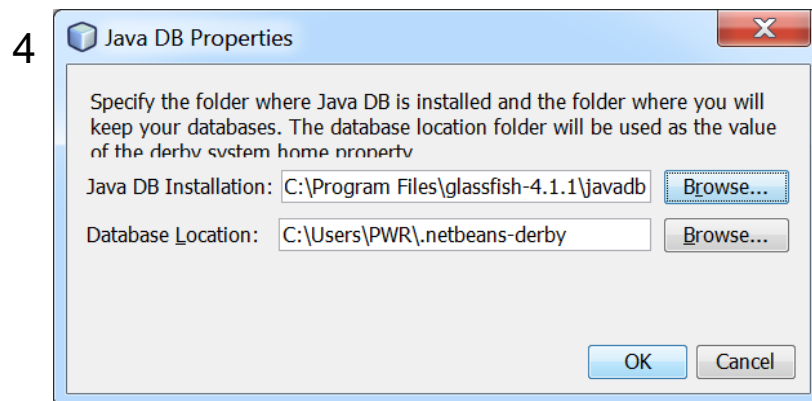
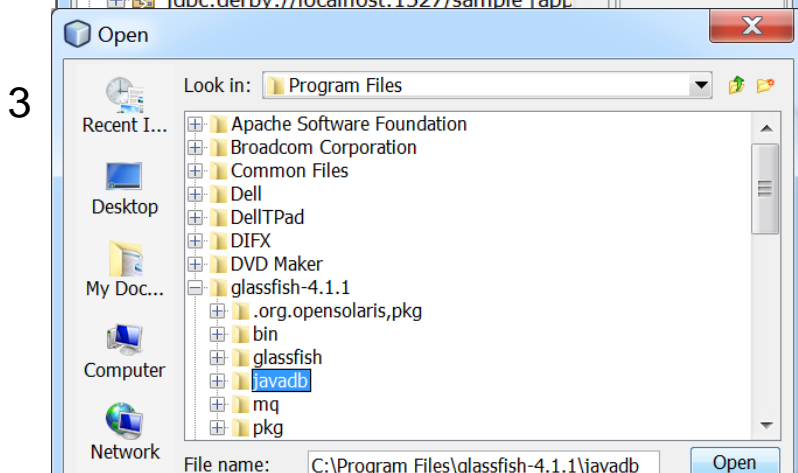
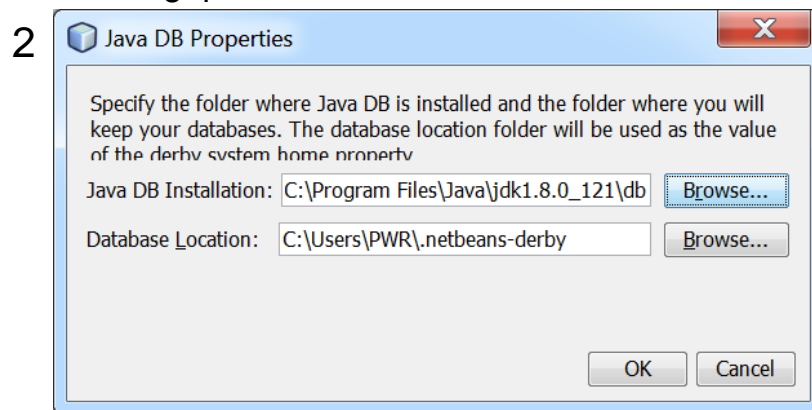
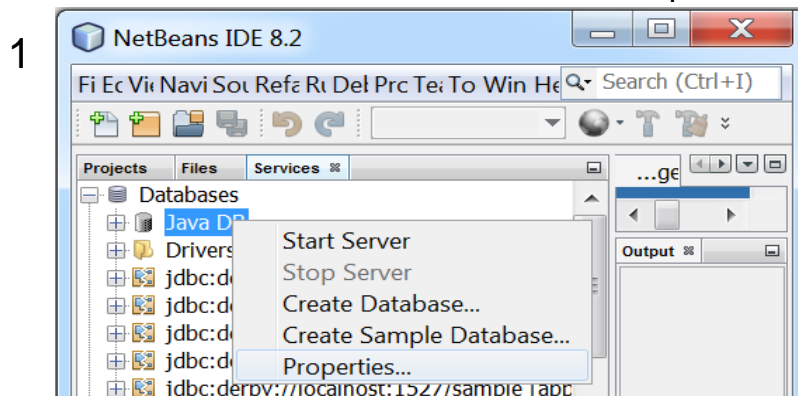
3. Należy wykonać kopię katalogu głównego, w którym znajdują się projekty wykonane podczas laboratorium 4. Jeśli te programy są w różnych katalogach, należy je skopiować do jednego katalogu, a po uruchomieniu spróbować przywrócić powiązania pomiędzy projektami **wg p.1.3 z instrukcji do laboratorium 3**, usuwając dotychczasowe powiązania np. za pomocą pozycji **Remove** wybierając katalog **Libraries** projektu i zaznaczając usuwany projekt:



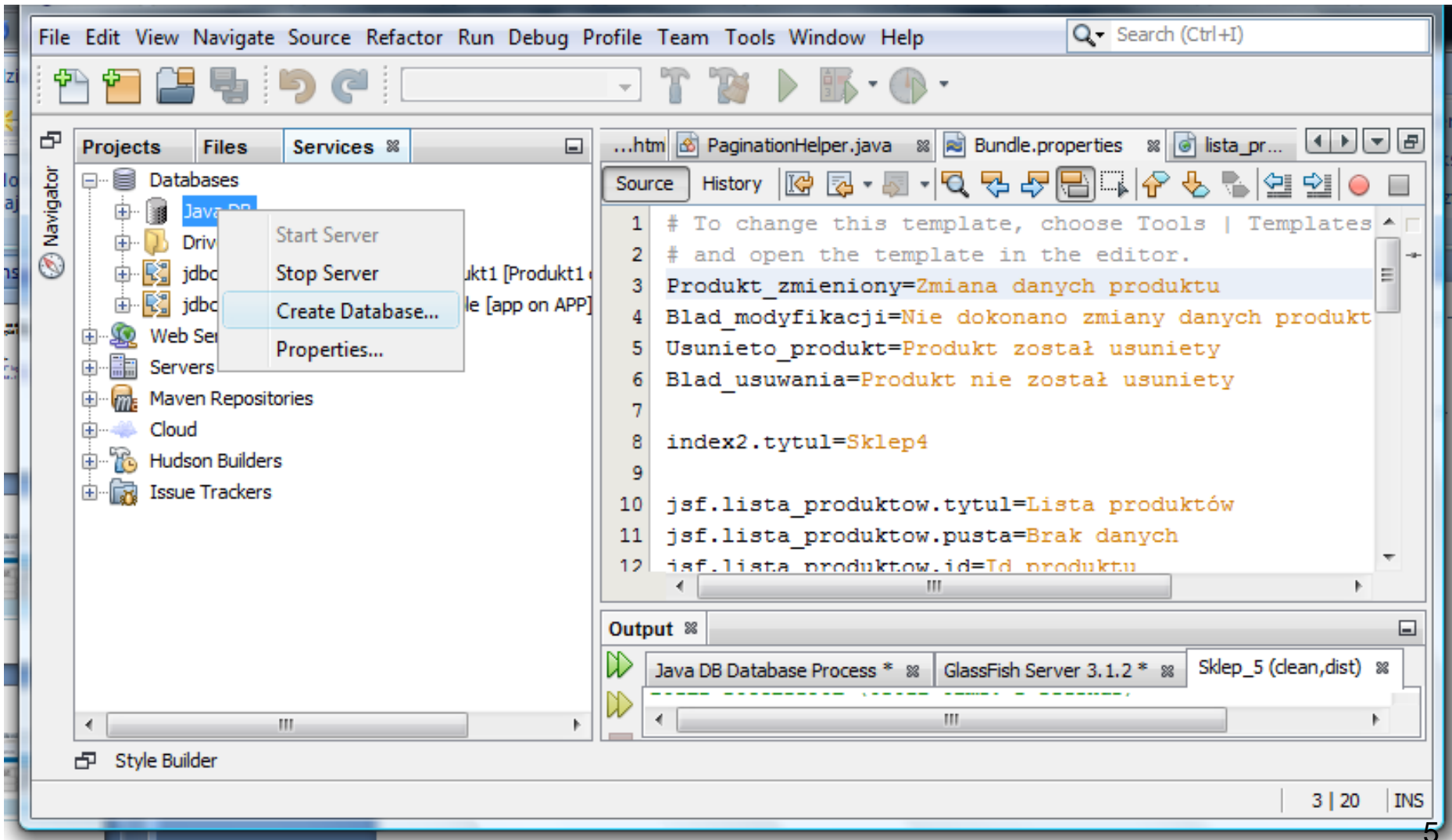
1(2). Należy uruchomić serwer bazy danych. W tym celu należy wykonać czynności pokazane poniżej w okienku Services. W p. 1(3) pokazano, jak zmienić program Java DB, gdy wystąpią problemy uruchomieniem serwera Java DB.



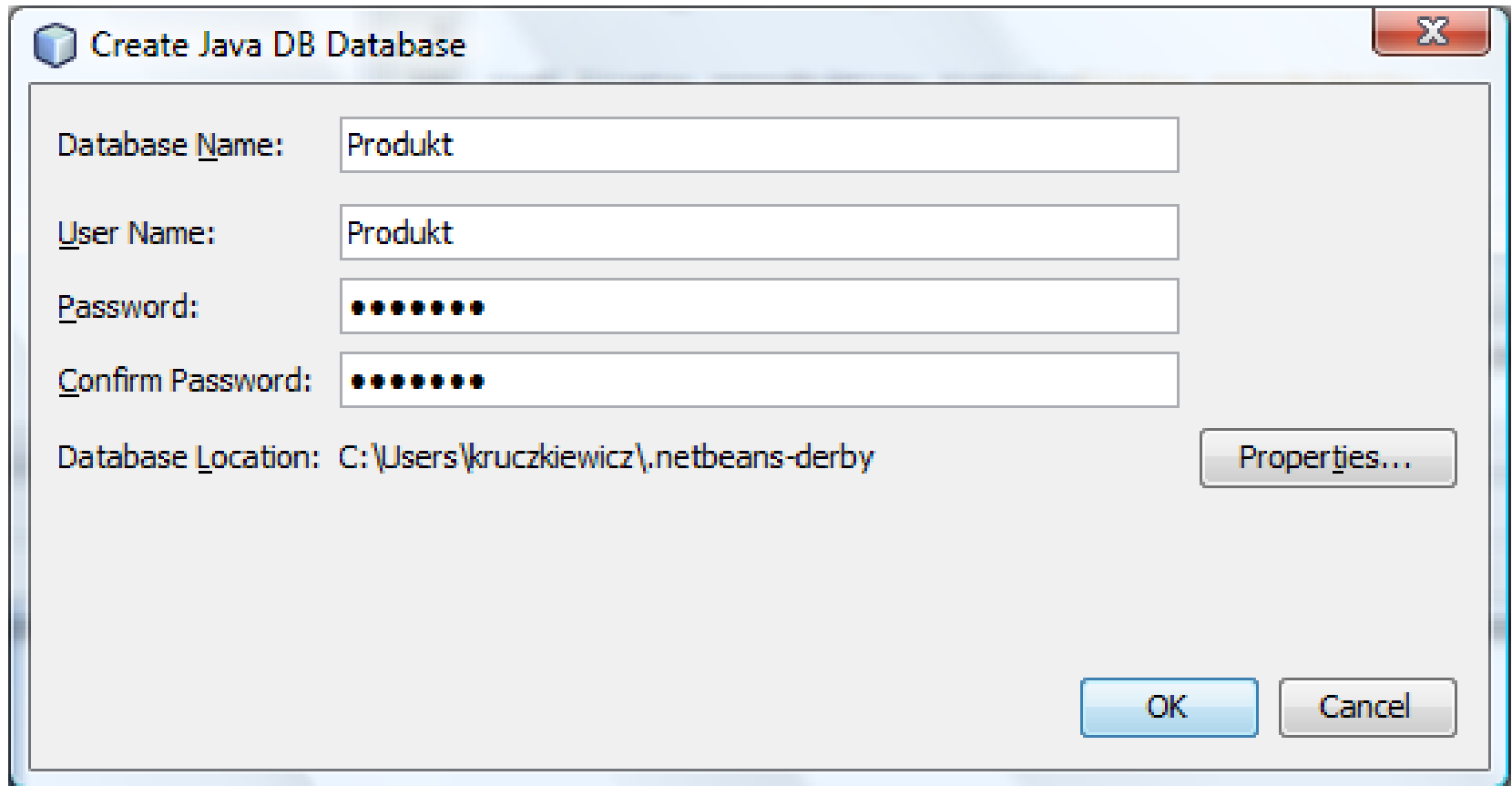
1(3). Jeśli serwer **Java DB** nie startuje, należy zmienić jego program pobrany z zainstalowanego serwera glassfish 4.1.1 i ponownie uruchomić wg p. 1a.



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



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



Create Java DB Database

Database Name: Produkt

User Name: Produkt

Password: ●●●●●●●●

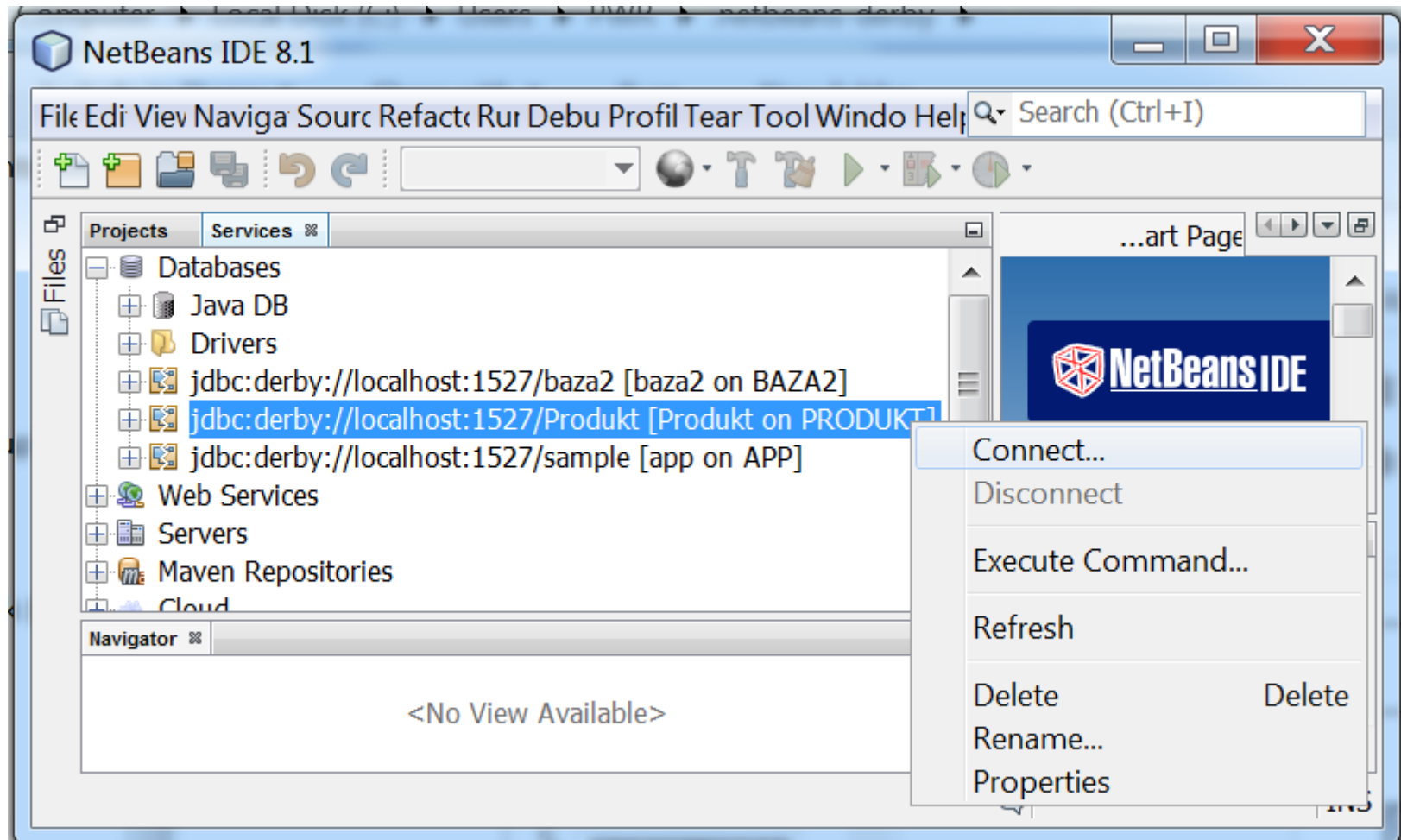
Confirm Password: ●●●●●●●●

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

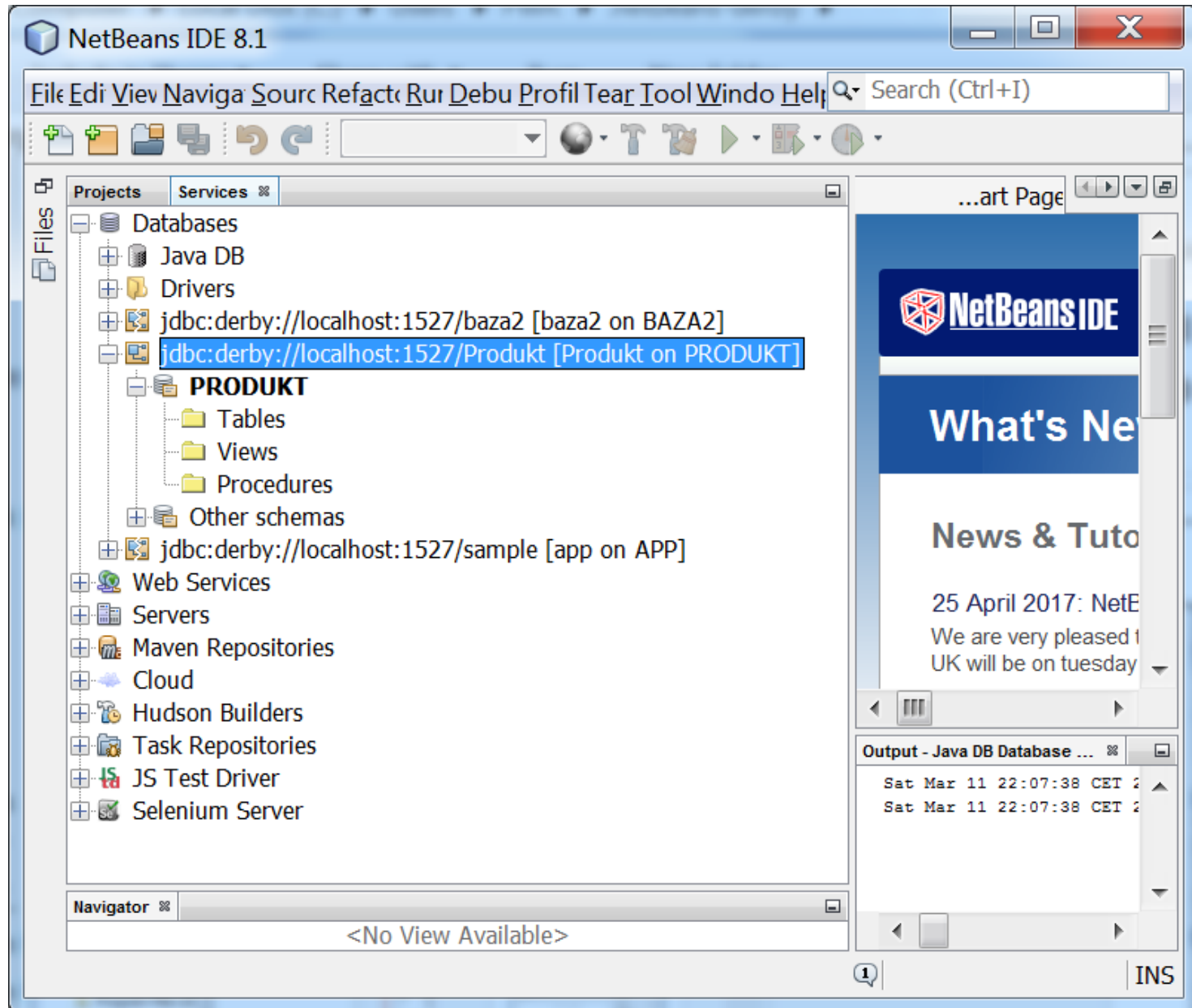
Properties...

OK Cancel

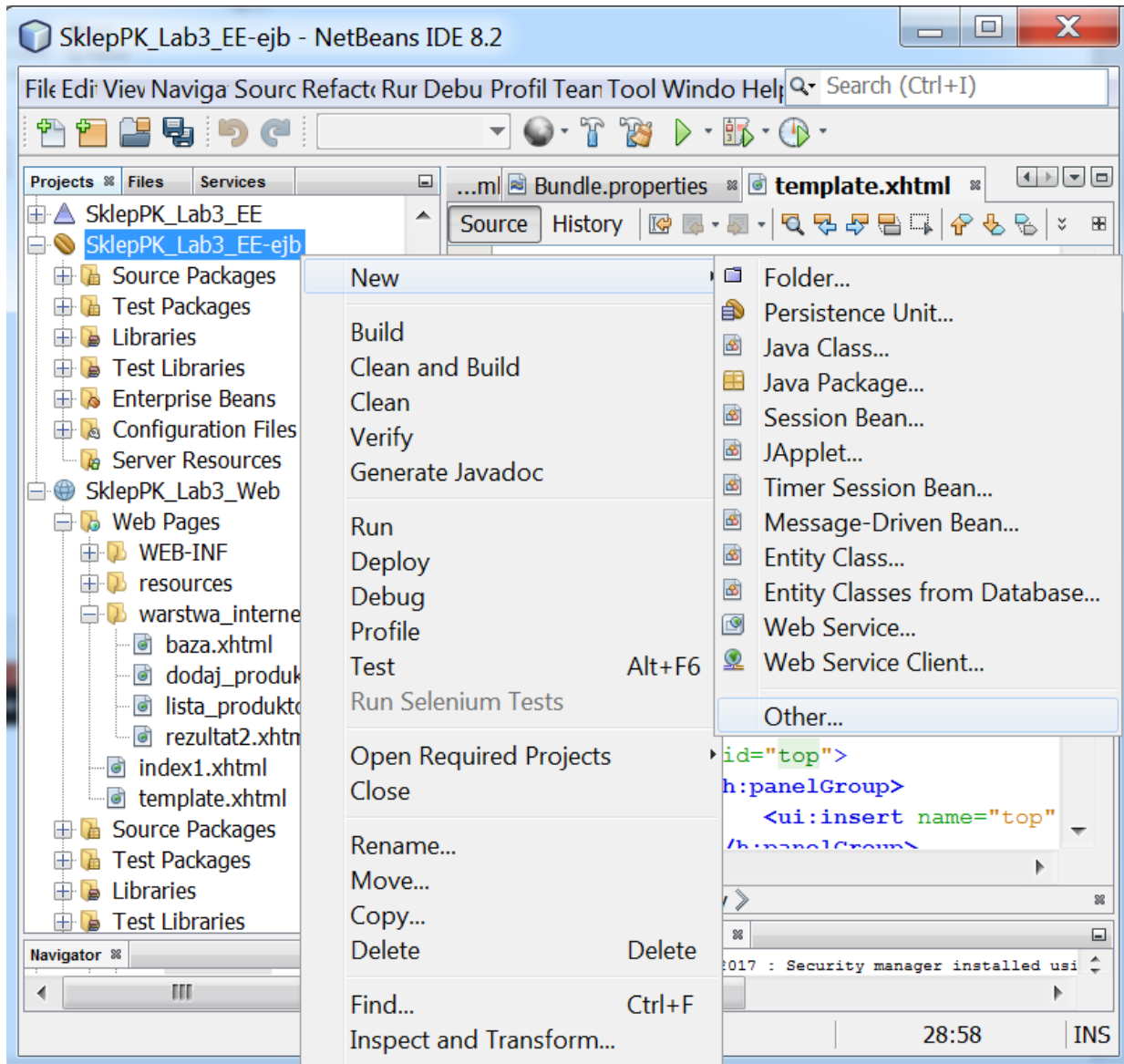
2.(3). Połączenie z pustą bazą danych (wybór pozycji **Connect** po kliknięciu prawym klawiszem myszy na połączeniu do bazy danych)



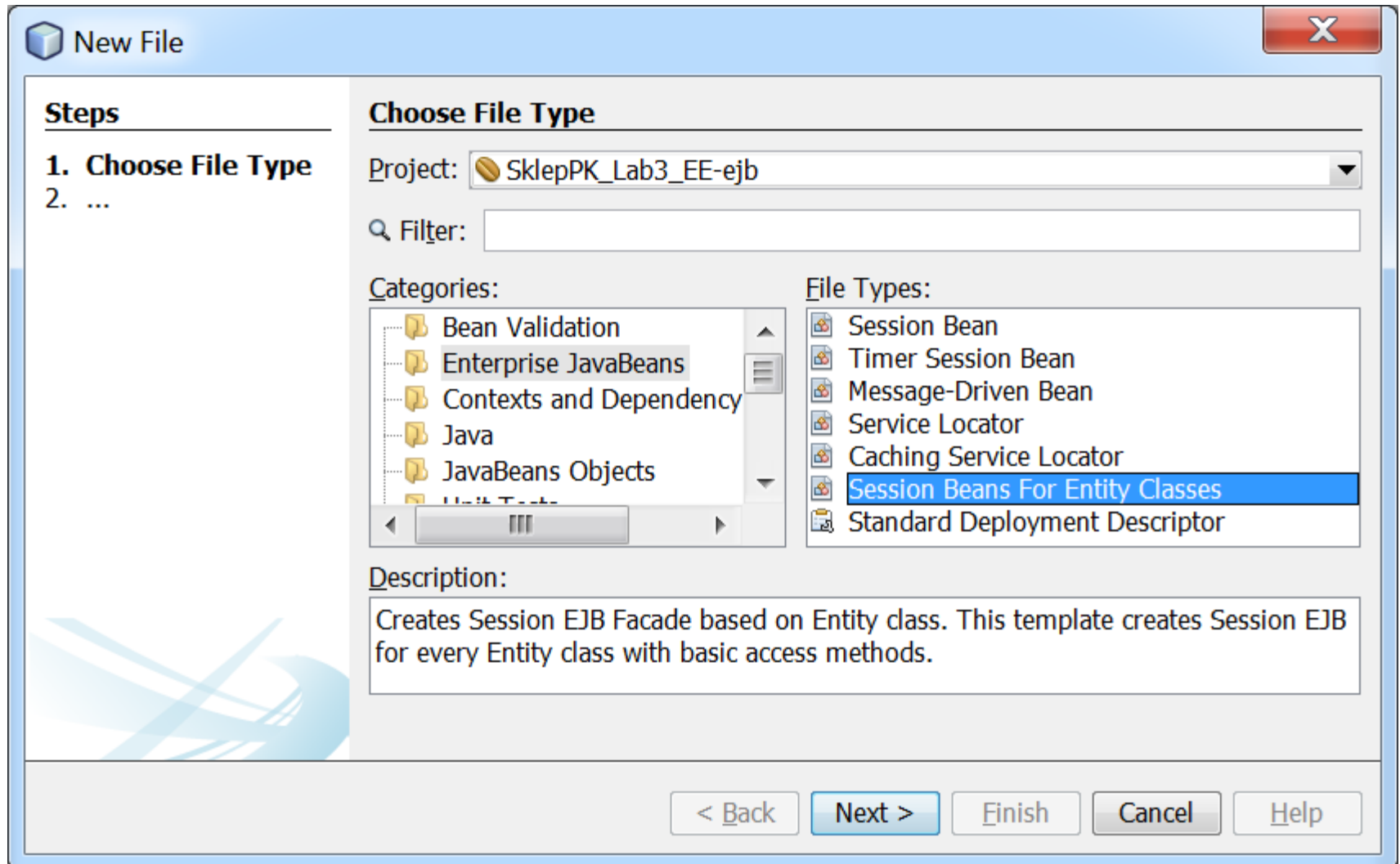
2.(4). Widok utworzonej pustej bazy danych **Produkt**



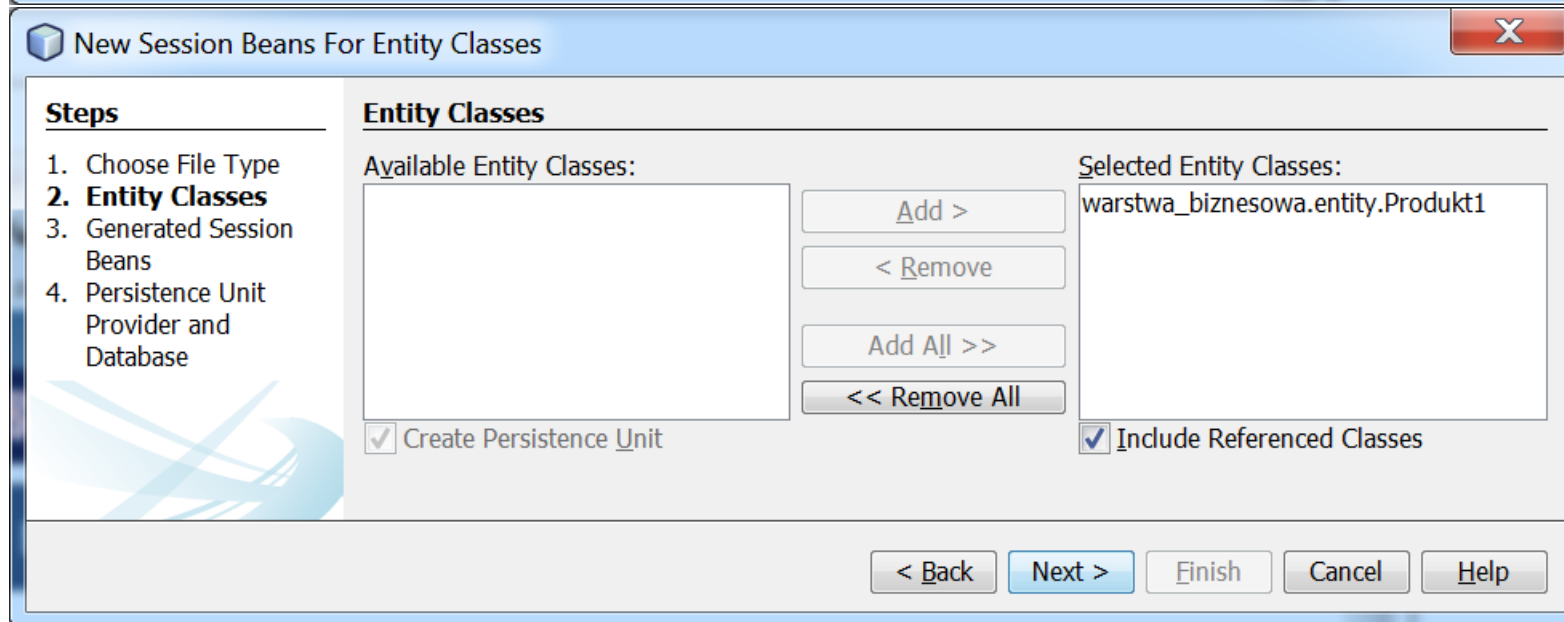
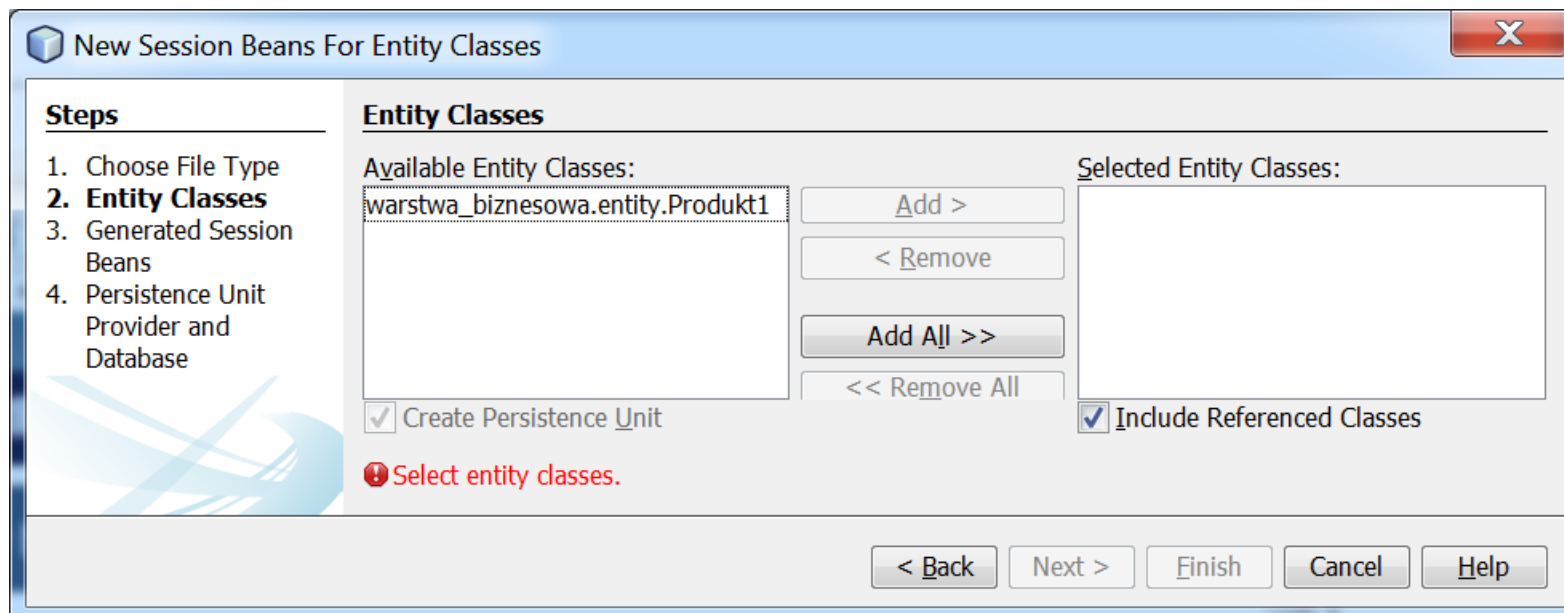
3.(1). Dodanie komponentów do utrwalania obiektów typu Produkt1 – **New/Other**



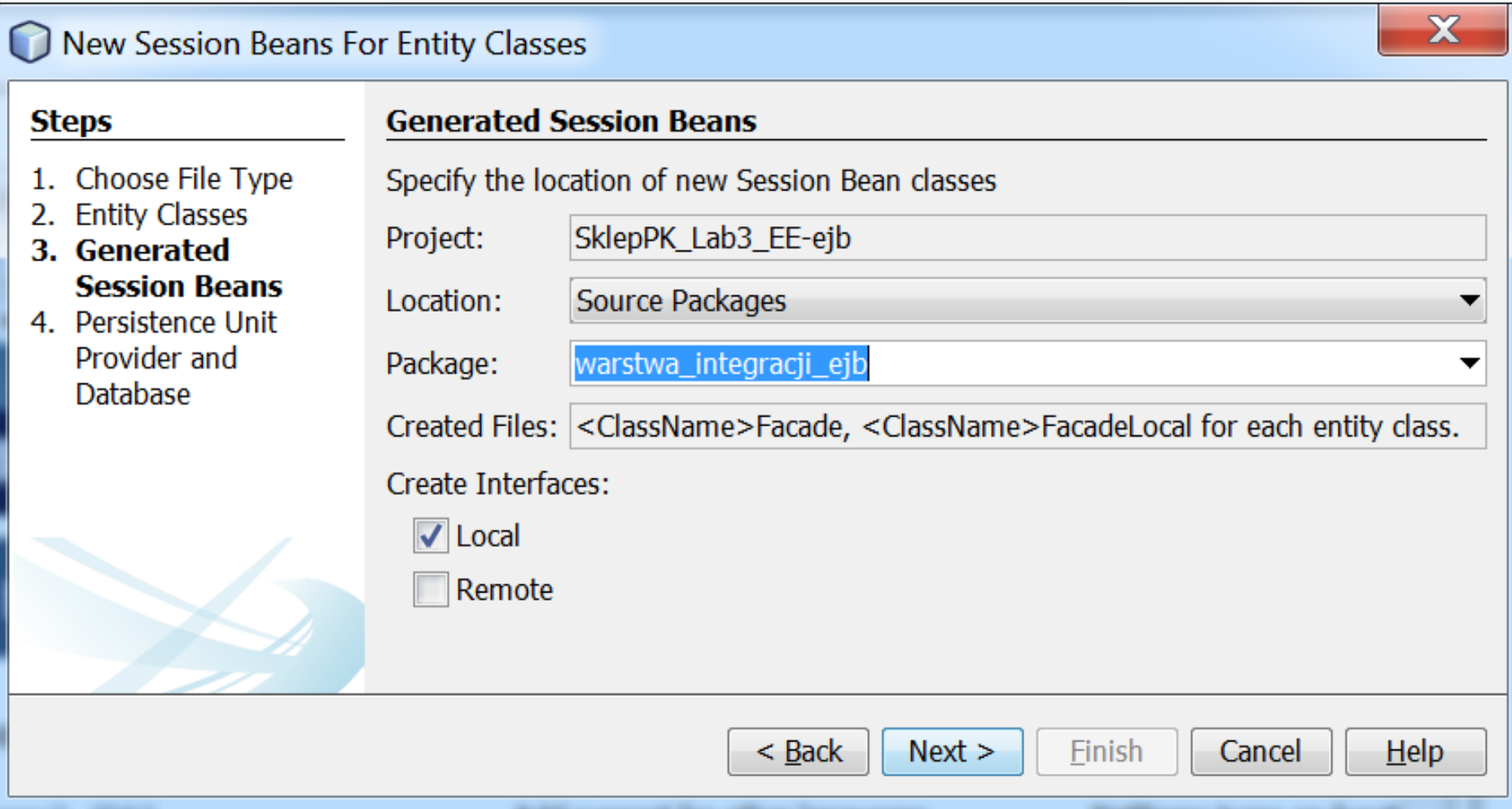
3.(2). Utworzenie pliku wspierającego ORM klasy Produkt – **Enterprise JavaBeans/SessionBeans for Entity Classes**



3.(3). Wybór klasy **Produkt1** typu **Entity** jako komponentu do utrwalania



3.(4). 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.

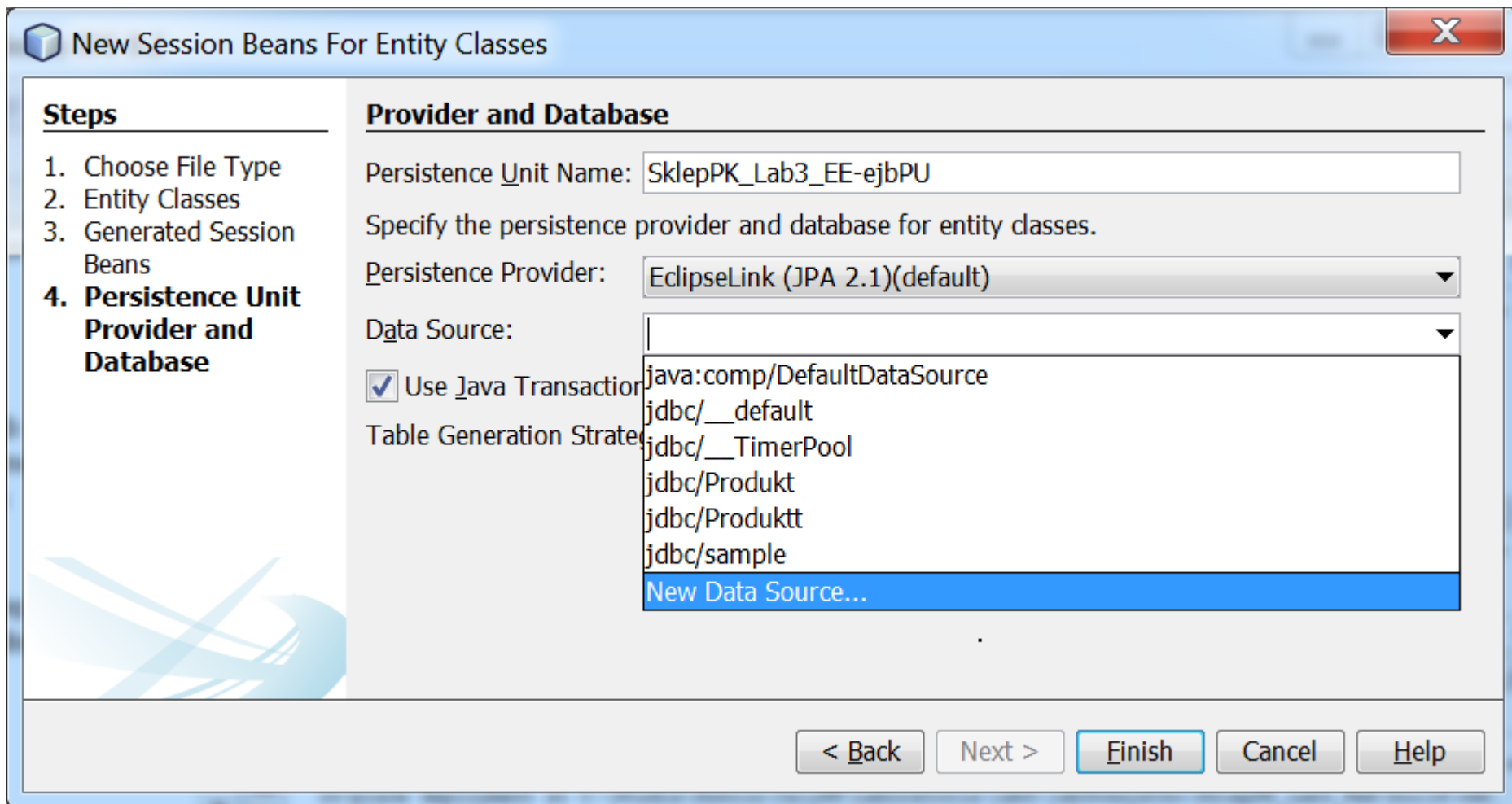
Create Interfaces:

Local

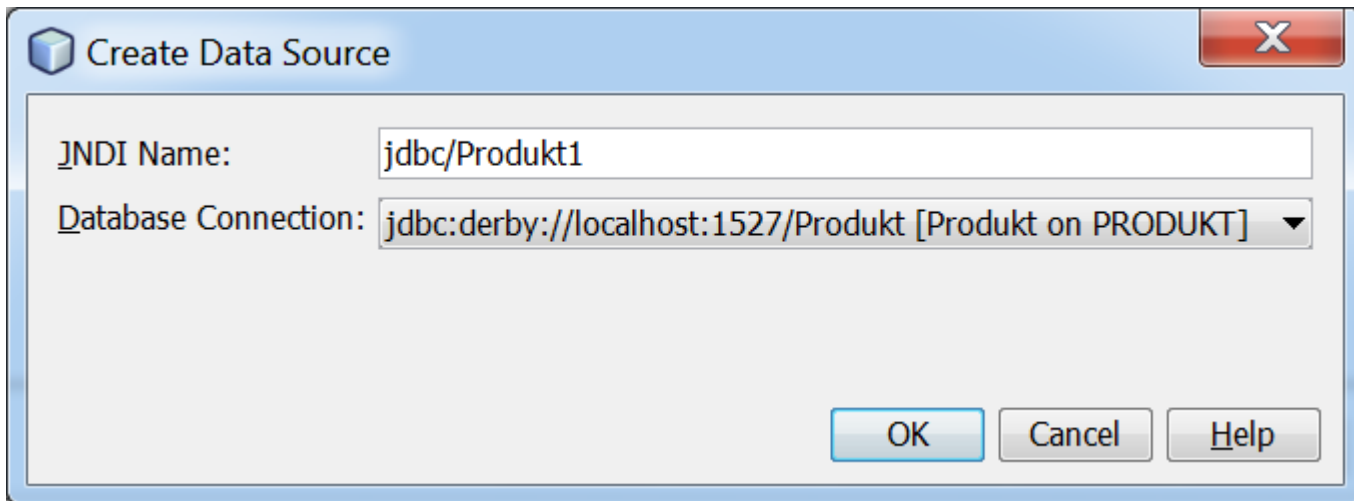
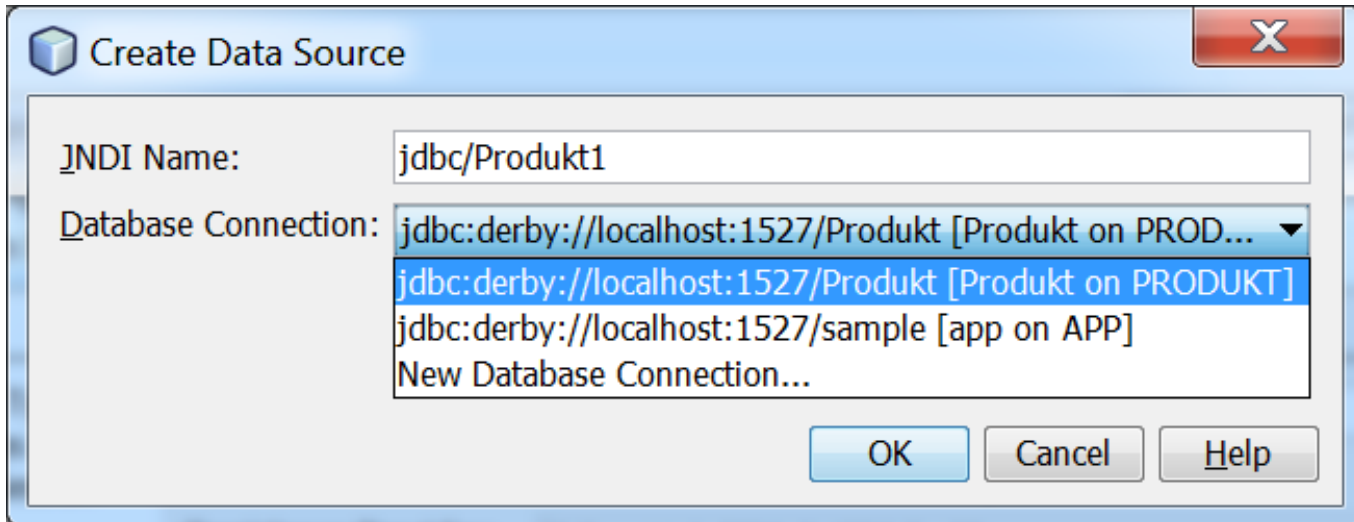
Remote

< Back Next > Finish Cancel Help

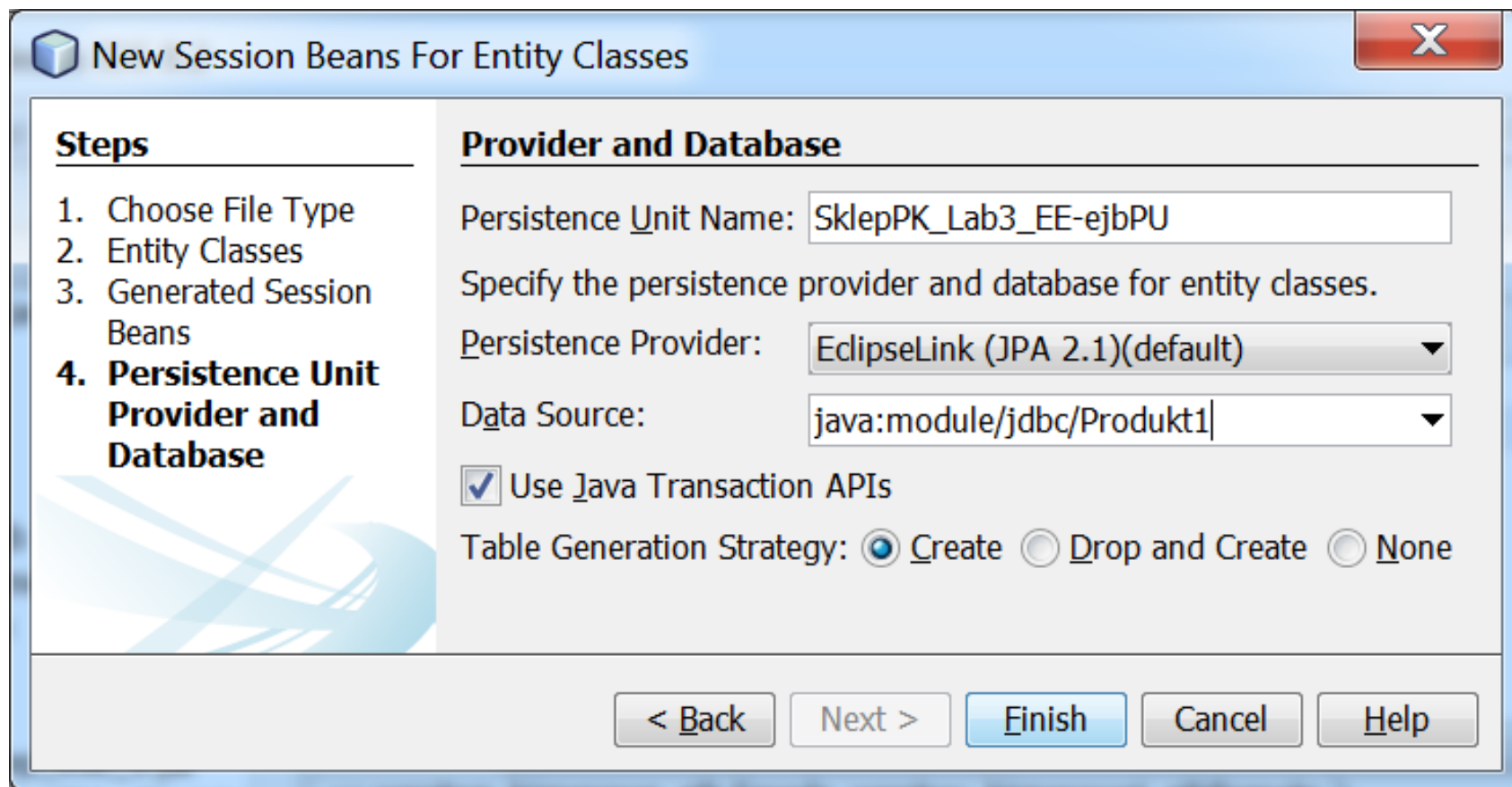
3. (5). Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) - utworzenie połączenia z bazą danych (wykonaną w kroku 2) za pomocą pozycji **New Data Source...**



3. (6). 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**



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



The screenshot shows the 'New Session Beans For Entity Classes' wizard in Eclipse IDE. The window title is 'New Session Beans For Entity Classes'. The 'Steps' pane on the left lists four steps: 1. Choose File Type, 2. Entity Classes, 3. Generated Session Beans, and 4. Persistence Unit Provider and Database (which is currently selected). The main area is titled 'Provider and Database' and contains the following fields and options:

- 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

At the bottom of the wizard, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted in blue.

3. (8). Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) - jawny wybór klasy **Produkt1** jako **Entity Class** – usunięcie wyboru opcji **Include All Entity Classes in „SklepPK_Lab3_EE-ejb” Module”**

The screenshot displays the NetBeans IDE 8.2 interface. The main window shows the configuration for a persistence.xml file. The 'General' tab is selected, and the following settings are visible:

- Persistence Unit Name: SklepPK_Lab3_EE-ejbPU
- 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
- Validation Strategy: Auto Callback None
- Shared Cache Mode: All None Enable Selective Disable Selective Unspecified
- Include All Entity Classes in "SklepPK_Lab3_EE-ejb" Module
- Include Entity Classes: (empty field with 'Add Class...' and 'Remove' buttons)

The project tree on the left shows the following structure:

- Sklep_6SE_1
 - Sklep_GUIPK_lab3_EE_Desktop
 - SklepPK_interfejs_1
 - 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
 - META-INF
 - glassfish-resources.xml
 - MANIFEST.MF
 - persistence.xml
 - Server Resources
- SklepPK_Lab3_Web

The bottom panel shows search results and output:

- Search Results: Java DB Database Process, GlassFish Server 4.1.1, Sklep_GUIPK_lab3_EE_Desktop (run)
- Output: compile: library-inclusion-in-archive:

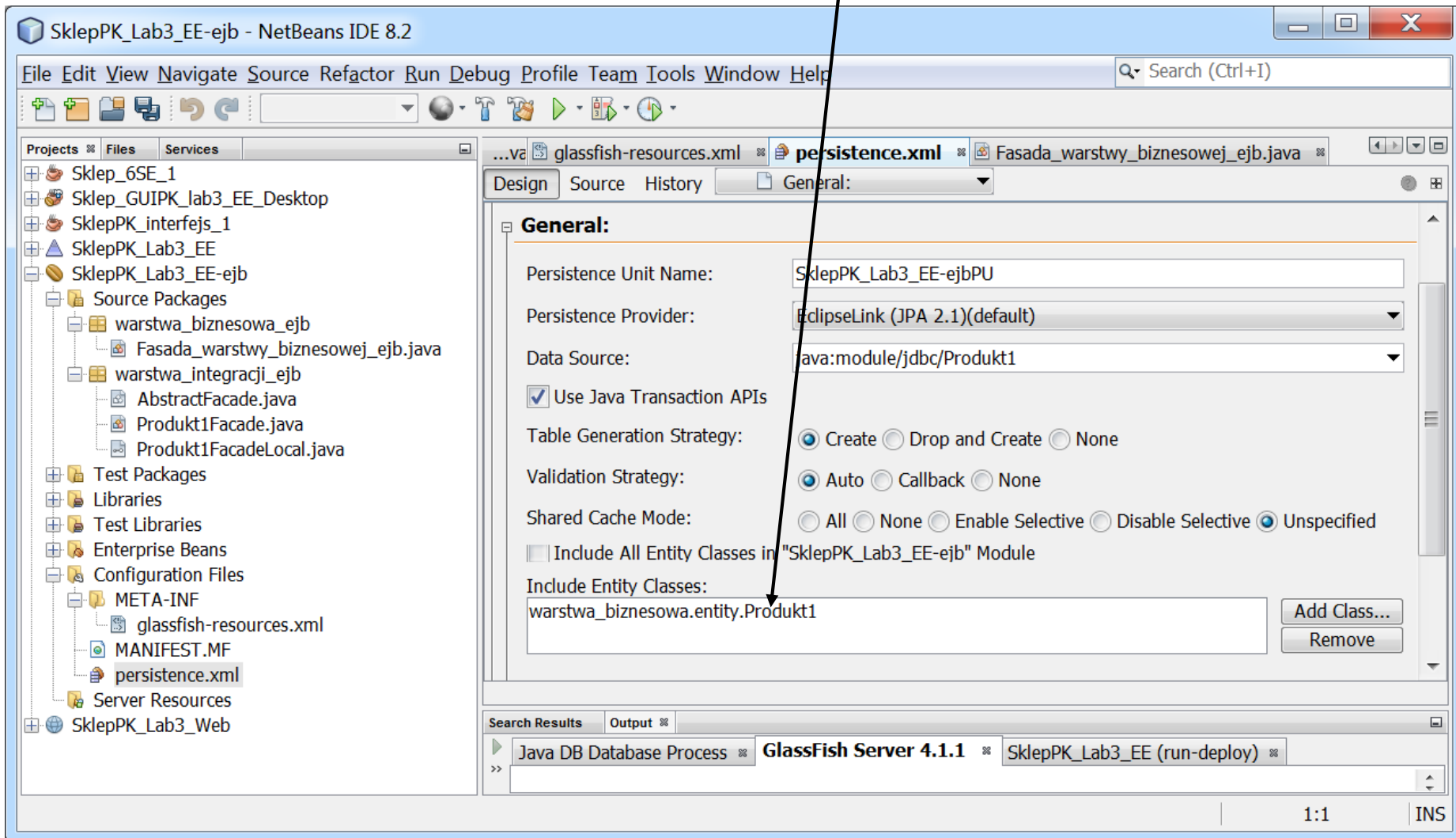
3. (9). Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) - jawny wybór klasy **Produkt1** jako **Entity Class**

The screenshot shows the NetBeans IDE 8.2 interface. The main window displays the configuration for a persistence.xml file. The 'General' tab is selected, showing the following settings:

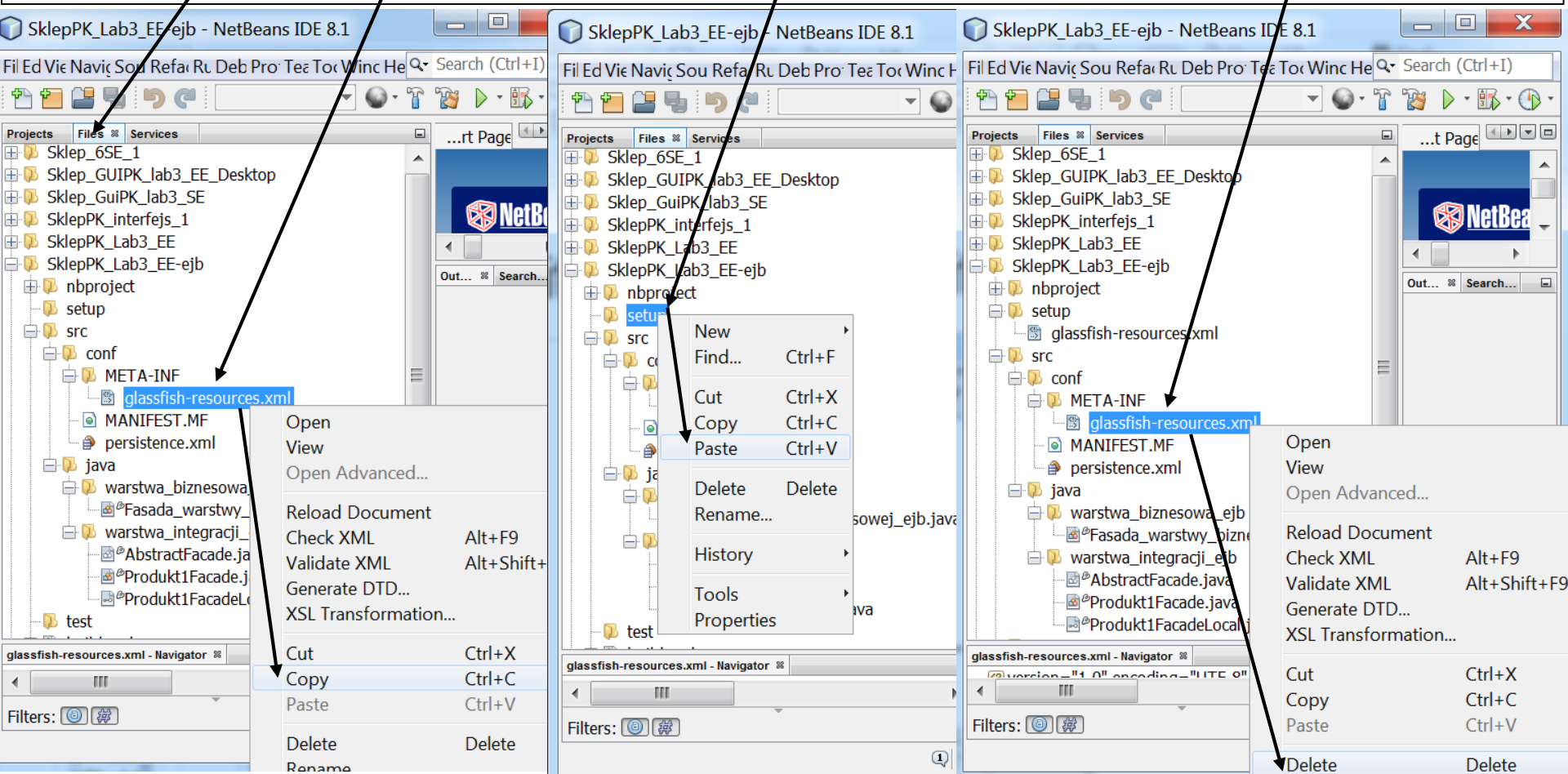
- Persistence Unit Name: SklepPK_Lab3_EE-ejbPU
- Persistence Provider: EclipseLink (JPA 2.1)(default)
- Data Source: java:module/jdbc/Produkt1
- Use Java Transaction APIs
- Table Generation Strategy: Create Drop and Create None
- Validation Strategy: Auto Callback None
- Shared Cache Mode: All None Enable Selective Disable Selective Unspecified
- Include All Entity Classes in "SklepPK_Lab3_EE-ejb" Module
- Include Entity Classes: (empty list)

The 'Add Class...' button is highlighted, and a dialog box titled 'Add Entity Class' is open, showing the class 'warstwa_biznesowa.entity.Produkt1' selected. The dialog box has 'OK', 'Cancel', and 'Help' buttons.

3. (10). Wykonanie pliku typu **persistence.xml** (deskryptora utrwalania JPA) – rezultat: jawnie wybrana klasa **Produkt1**



3. (11 cd) Inny sposób zmiany położenia wygenerowanego pliku **glassfish-resource.xml** w zakładce **Files** – przesunięcie do folderu **Server Resources** w projekcie. Jeśli brakuje folderu **setup**, należy go utworzyć w projekcie **SklepPK_Lab3_EE-ejb**



3. (12). Modyfikacja wygenerowanej nazwy JNDI `java:module/jdbc/Produkt1` na `jdbc/Produkt1` w przemieszczonym pliku `glassfish-resources.xml`

The screenshot shows the NetBeans IDE 8.2 interface. The left sidebar displays the project structure for 'SklepPK_Lab3_EE-ejb', with 'glassfish-resources.xml' selected under 'Server Resources'. The main editor window shows the XML content of 'glassfish-resources.xml'. The XML is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Se
<resources>
  <jdbc-connection-pool allow-non-component-callers="false" associate-wi
    <property name="serverName" value="localhost"/>
    <property name="portNumber" value="1527"/>
    <property name="databaseName" value="Produkt"/>
    <property name="User" value="Produkt"/>
    <property name="Password" value="Produkt"/>
    <property name="URL" value="jdbc:derby://localhost:1527/Produkt"/>
    <property name="driverClass" value="org.apache.derby.jdbc.ClientDr
  </jdbc-connection-pool>
  <jdbc-resource enabled="true" jndi-name="java:module/jdbc/Produkt1"
    object-type="user"
    pool-name="derby_net_Produkt_ProduktPool"/>
</resources>
```

An arrow points from the text in the title to the `jndi-name="java:module/jdbc/Produkt1"` attribute in the XML. The bottom status bar shows the time as 14:20 and the mode as INS.

3. (14). Modyfikacja wygenerowanej nazwy JNDI **java:module/jdbc/Produkt1** na **jdbc/Produkt1** w drugim wygenerowanym pliku **persistence.xml**

The screenshot displays the NetBeans IDE 8.2 interface. The main window shows the configuration for the persistence.xml file. The 'Data Source' field is set to 'jdbc/Produkt1'. The 'Include Entity Classes' field contains 'warstwa_biznesowa.entity.Produkt1'. The 'Search Results' window shows the output of the deployment process, indicating a successful build.

General:

- Persistence Unit Name: SklepPK_Lab3_EE-ejbPU
- Persistence Provider: EclipseLink (JPA 2.1)(default)
- Data Source: jdbc/Produkt1
- Use Java Transaction APIs
- Table Generation Strategy: Create Drop and Create None
- Validation Strategy: Auto Callback None
- Shared Cache Mode: All None Enable Selective Disable Selective Unspecified
- Include All Entity Classes in "SklepPK_Lab3_EE-ejb" Module
- Include Entity Classes: warstwa_biznesowa.entity.Produkt1

Search Results

```
Java DB Database Process  GlassFish Server 4.1.1  SklepPK_Lab3_EE (run-deploy)
pre-run-deploy:
Initial deploying SklepPK_Lab3_EE to C:\Studia\Szkola\CalyPK\Laboratoria\lab5\lab3wszystko\SklepPK_Lab3_EE\dis
Completed initial distribution of SklepPK_Lab3_EE
post-run-deploy:
run-deploy:
BUILD SUCCESSFUL (total time: 2 seconds)
```

3.(15). Widok komponentu **Session Bean fo Entity Classes (JPA)** **Produkt1Facade**, który implementuje interfejs **Produkt1FacadeLocal** i dziedziczy po klasie abstrakcyjnej **AbstractFacade**

The screenshot displays the NetBeans IDE 8.2 interface. The left sidebar shows a project tree for 'Sklep_6SE_1' with a sub-package 'warstwa_integracji_ejb' containing 'Produkt1Facade.java'. The main editor window shows the source code of 'Produkt1Facade.java'.

```
1 package warstwa_integracji_ejb;
2
3 import javax.ejb.Stateless;
4 import javax.persistence.EntityManager;
5 import javax.persistence.PersistenceContext;
6 import warstwa_biznesowa.entity.Produkt1;
7
8 @Stateless
9 public class Produkt1Facade extends AbstractFacade<Produkt1>
10     implements Produkt1FacadeLocal {
11
12     @PersistenceContext(unitName = "SklepPK_Lab3_EE-ejbPU")
13     private EntityManager em;
14
15     @Override
16     protected EntityManager getEntityManager() {
17         return em;
18     }
19     public Produkt1Facade() {
20         super(Produkt1.class);
21     }
22 }
```

The IDE also shows a 'Output' window at the bottom with the text 'Java DB Database Process' and 'SklepPK_Lab3_EE-ejb (clean)'. The system tray at the bottom right shows the time '10:29' and the window state 'INS'.

3.(16). Widok wygenerowanego interfejsu **Produkt1FacadeLocal**

The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows the source code for `Produkt1FacadeLocal.java`. The code defines a `@Local` interface with the following methods:

```
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 public interface Produkt1FacadeLocal {
18     void create(Produkt1 produkt1);
19     void edit(Produkt1 produkt1);
20     void remove(Produkt1 produkt1);
21     Produkt1 find(Object id);
22     List<Produkt1> findAll();
23     List<Produkt1> findRange(int[] range);
24     int count();
25 }
```

The left sidebar shows the project structure for `SklepPK_Lab3_EE-ejb`, with `Produkt1FacadeLocal.java` selected under the `warstwa_integracji_ejb` package. The bottom status bar shows the time `24:17` and the text `INS`.

3.(17). Wygenerowana klasa abstrakcyjna **AbstractFacade** (część 1)

The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows the source code for the `AbstractFacade` class, which is part of the `warstwa_integracji_ejb` package. The code defines an abstract class with a private `entityClass` field and several abstract methods: `getEntityManager()`, `create()`, `edit()`, and `remove()`. The `remove()` method is currently selected and highlighted in blue. The IDE's left sidebar shows the project structure, including the `warstwa_integracji_ejb` package and its sub-packages. The bottom status bar indicates that the Java DB Database Process is running for the `SklepPK_Lab3_EE-ejb` project.

```
1 package warstwa_integracji_ejb;
2
3 import java.util.List;
4 import javax.persistence.EntityManager;
5
6 public abstract class AbstractFacade<T> {
7
8     private Class<T> entityClass;
9
10    public AbstractFacade(Class<T> entityClass) {
11        this.entityClass = entityClass;
12    }
13
14    protected abstract EntityManager getEntityManager();
15
16    public void create(T entity) {
17        getEntityManager().persist(entity);
18    }
19
20    public void edit(T entity) {
21        getEntityManager().merge(entity);
22    }
23
24    public void remove(T entity) {
25        getEntityManager().remove(getEntityManager().merge(entity));
26    }
27 }
```

3.(18). Wygenerowana klasa abstrakcyjna **AbstractFacade** (część 2)

The screenshot displays the NetBeans IDE 8.2 environment. The main editor window shows the source code for the `AbstractFacade` class, which is part of the `warstwa_integracji_ejb` package. The code defines three methods:

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

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar, and a project explorer on the left. The project explorer shows the following structure:

- SklepPK_Lab3_EE-ejb
- Java EE Modules
 - SklepPK_Lab3_EE-ejb.jar
- Configuration Files
- Server Resources
- Source Packages
 - warstwa_biznesowa_ejb
 - warstwa_integracji_ejb
 - AbstractFacade.java
 - Produkt1Facade.java
 - Produkt1FacadeLocal.java
- Test Packages
- Libraries
- Test Libraries
- Enterprise Beans
- Configuration Files
 - META-INF
 - MANIFEST.MF
 - persistence.xml
- Server Resources
 - glassfish-resources.xml
- SklepPK_Lab3_Web

The output console at the bottom shows the following message:

```
Java DB Database Process SklepPK_Lab3_EE-ejb (clean)
```

The status bar at the bottom right indicates the current position in the code: 50:6 INS.

4. (1). Przekazanie nadawania **id** klasie **Produkt1** typu **Entity** mechanizmowi 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(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;  
}
```

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;

4.(2). **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;  
}
```

4. (3). 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;  
}
```

4. (4). 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_);  
}
```


5. (1). Utworzenie referencji do komponentu **Produkt1Facade** w klasie **Fasada_warstwy_biznesowej_ejb**

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_produk2.xhtml

fasada - Navigator

Members

- remove(Produkt_dto p)
- setStan(boolean stan)
- utworz_produk(Produkt_dto produkt_dto)
- fasada : Fasada_warstwy_biznesowej

Source History

```
1 ...5 lines
6 package warstwa_biznesowa_ejb;
7
8 import java.util.ArrayList;
9 import javax.ejb.Stateless;
10 import warstwa_biznesowa.Fasada_warstwy_biznesowej;
11 import warstwa_biznesowa.dto.Produkt;
12
13 /**...4 lines */
17 @Stateless
18 public class Fasada_warstwy_biznesowej
19     implements Fasada_warstwy_biznesowej
20 {
21     Fasada_warstwy_biznesowej fasada;
22
23     public void utworz_produk(Produkt produkt)
24     {
25         fasada.utworz_produk(produkt);
26     }
27
28     public Produkt_dto dane_produk(Produkt produkt)
29     {
30         return fasada.dane_produk(produkt);
31     }
32 }
```

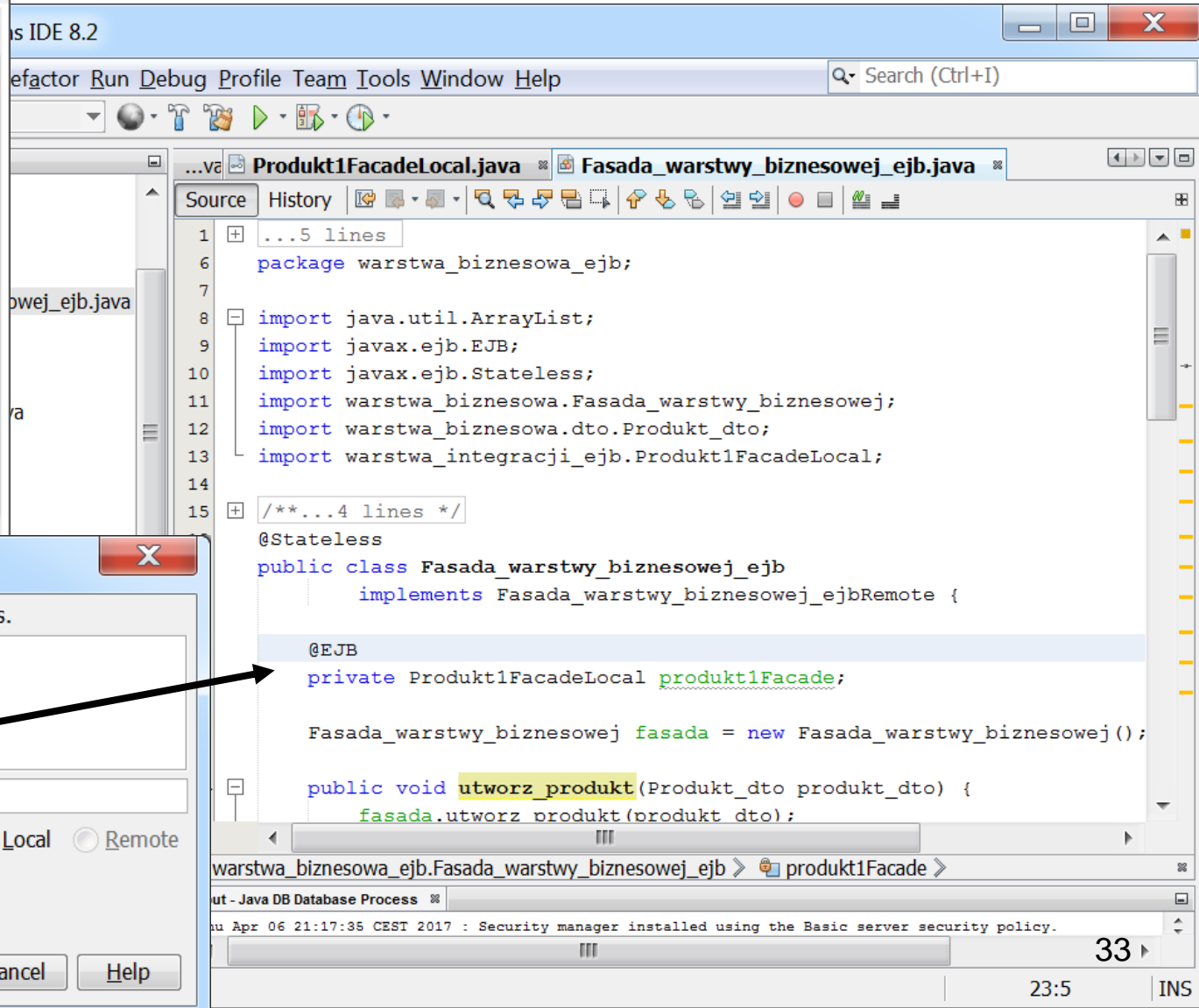
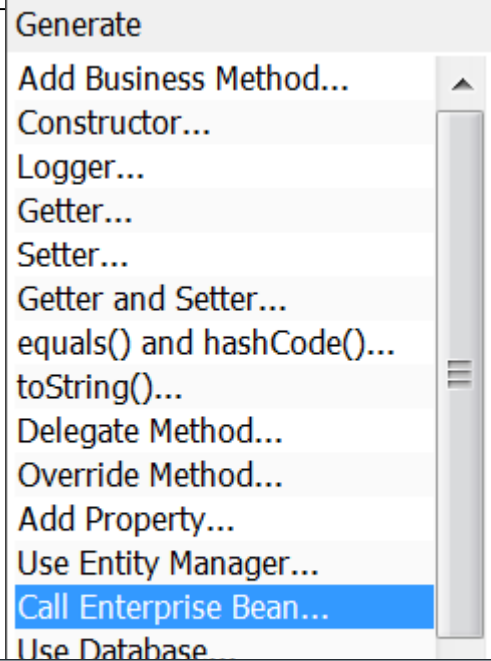
warstwa_biznesowa_ejb.Fasada_warstwy_biznesowej_ejb

Output - Java DB Database Process

Thu Apr 06 21:17:35 CEST 2017 : Security manager installed v

Navigate
Show Javadoc Alt+F1
Find Usages Alt+F7
Call Hierarchy
Insert Code... Alt+Insert
Fix Imports Ctrl+Shift+I
Refactor
Format Alt+Shift+F
Run File Shift+F6
Debug File Ctrl+Shift+F5
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Run Focused Test Method
Debug Focused Test Method
Run Into Method
New Watch... Ctrl+Shift+F7
Toggle Line Breakpoint Ctrl+F8
Profile

5.(2). 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**



5. (3). **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);  
        }  
    }  
}
```

5.(4). 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 {  
        pobierz() ;  
    } catch (Exception e) { }  
}
```

6. 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();
}
```

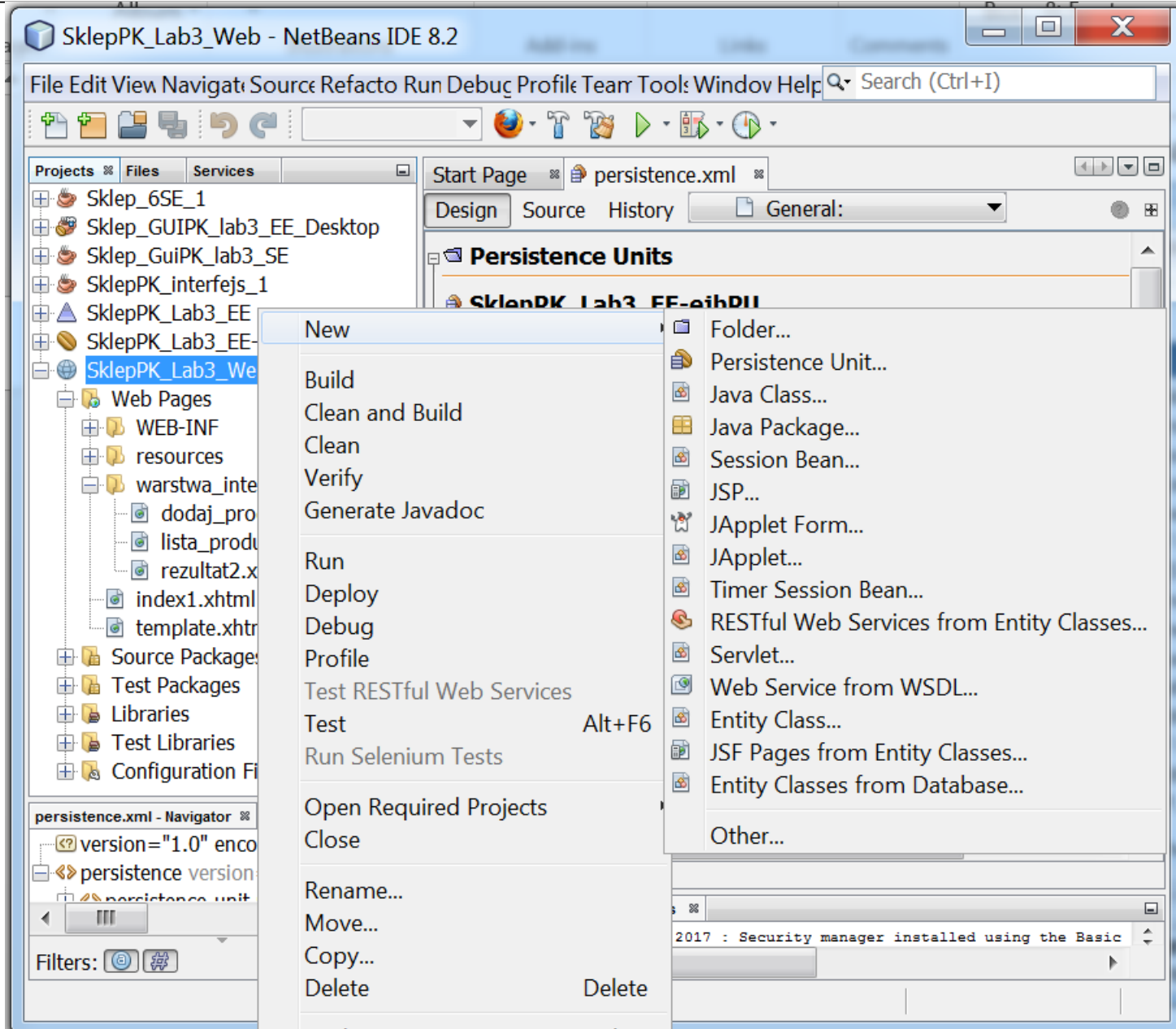
7. Dodane metody **zapisz()**, **pobierz()** w klasie **Managed_produk**t, wywołujące metody **zapisz()**, **pobierz()** z komponentu EJB

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

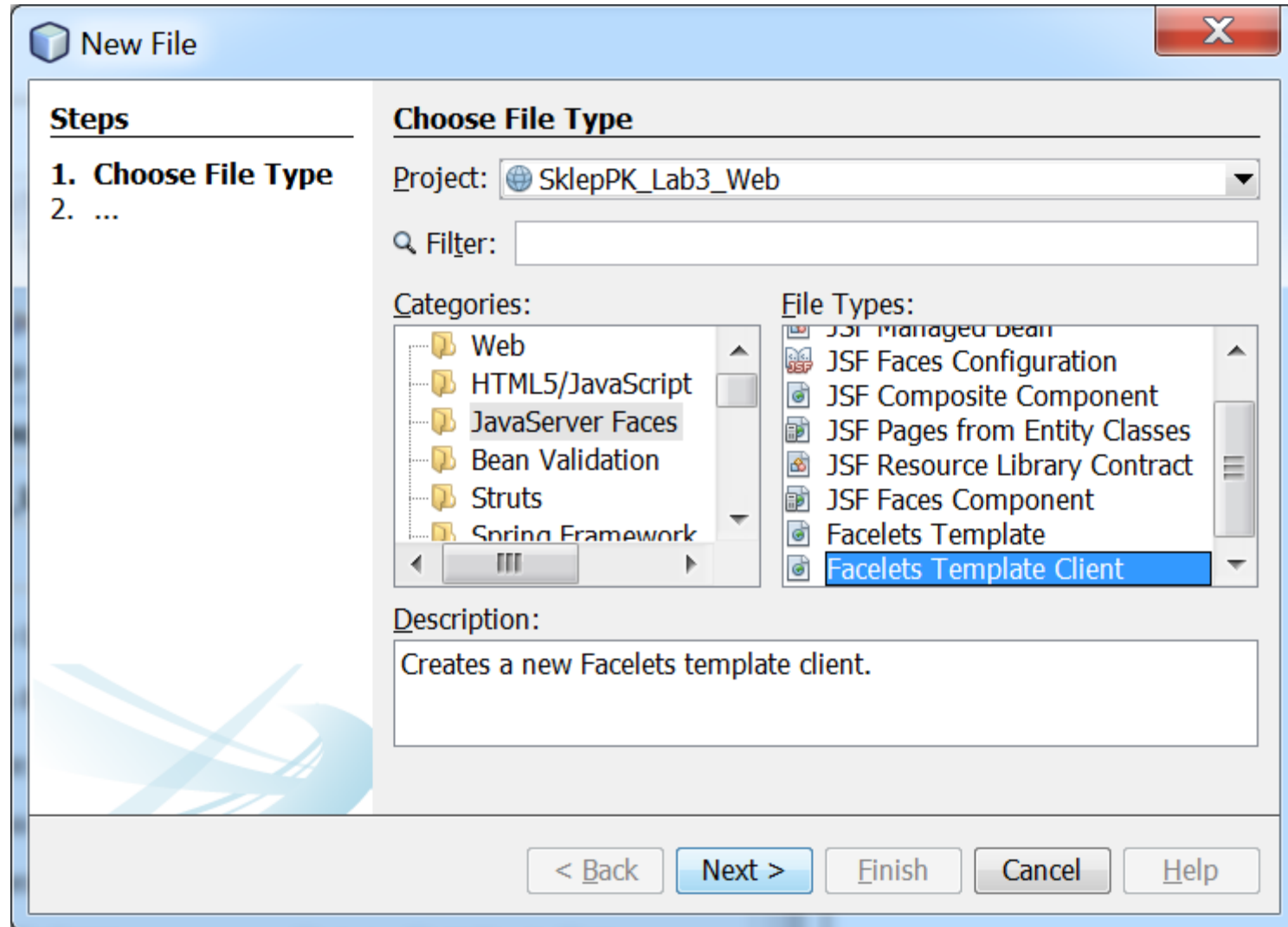
```
public String pobierz() {  
    fasada.pobierz();  
    refresh();  
    return "/faces/index1";  
}
```

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

8.(1). Dodanie nowej strony do obsługi bazy danych – **New/Other**



8.(2). Dodanie strony opartej na szablonie **template.xml**



8.(3). Dodanie strony **baza.xhtml** opartej na szablonie **template.xml**

New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: baza

Project: SklepPK_Lab3_Web

Folder:

Created File: ko_2\lab3wszystko\SklepPK_Lab3_Web\web\baza.xhtml

Template:

Generated Root Tag: <html>
 <ui:composition>

Sections To Generate:

Select a template for which the client will be generated.

Browse Folders

Folders:

- SklepPK_Lab3_Web - Web Pages
 - resources
 - warstwa_internetowa_jsf**
 - WEB-INF

New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: baza

Project: SklepPK_Lab3_Web

Folder: warstwa_internetowa_jsf

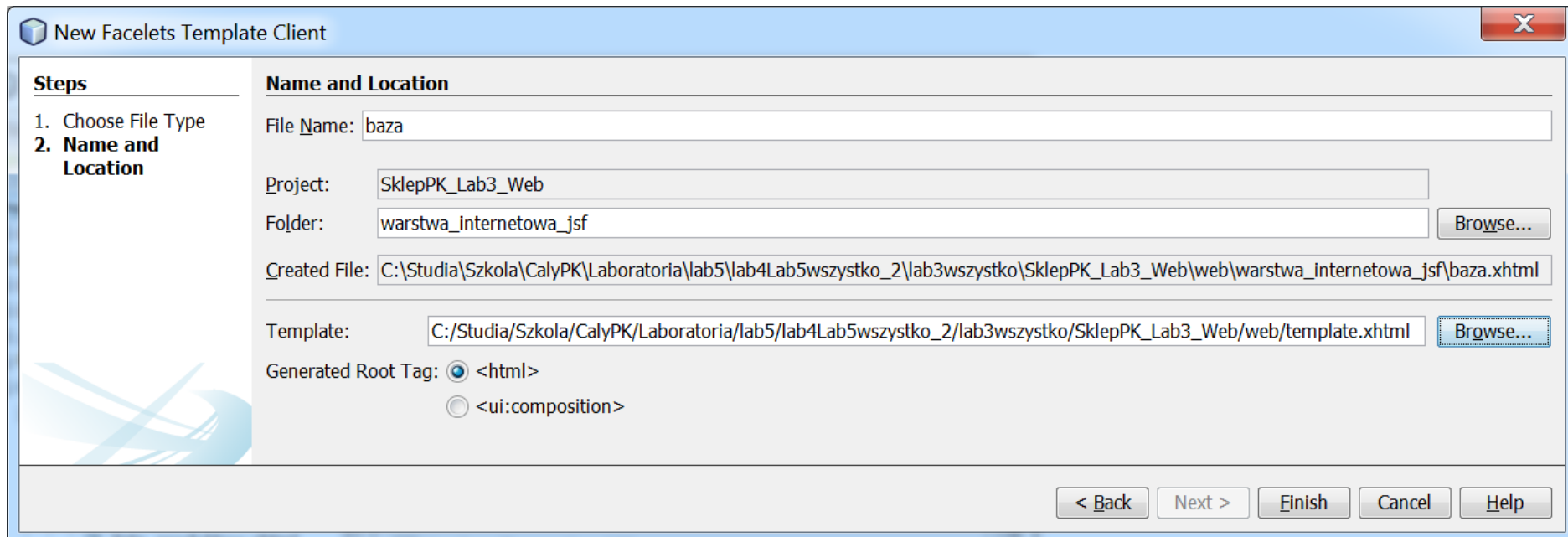
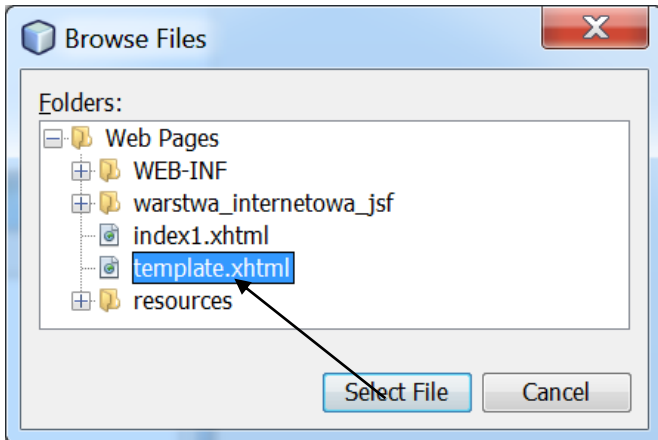
Created File: C:\Studia\Szkola\CalyPK\Laboratoria\lab5\lab4Lab5wszystko_2\lab3wszystko\SklepPK_Lab3_Web\web\warstwa_internetowa_jsf\baza.xhtml

Template:

Generated Root Tag: <html>
 <ui:composition>

Select a template for which the client will be generated.

8.(4). Dodanie strony **baza.xhtml** opartej na szablonie **template.xml**



8.(5). 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>
```

8.(6). 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

8.(7). Uzupełnienie pliku **Bundle.properties**

baza.tytul=Utrwalanie danych

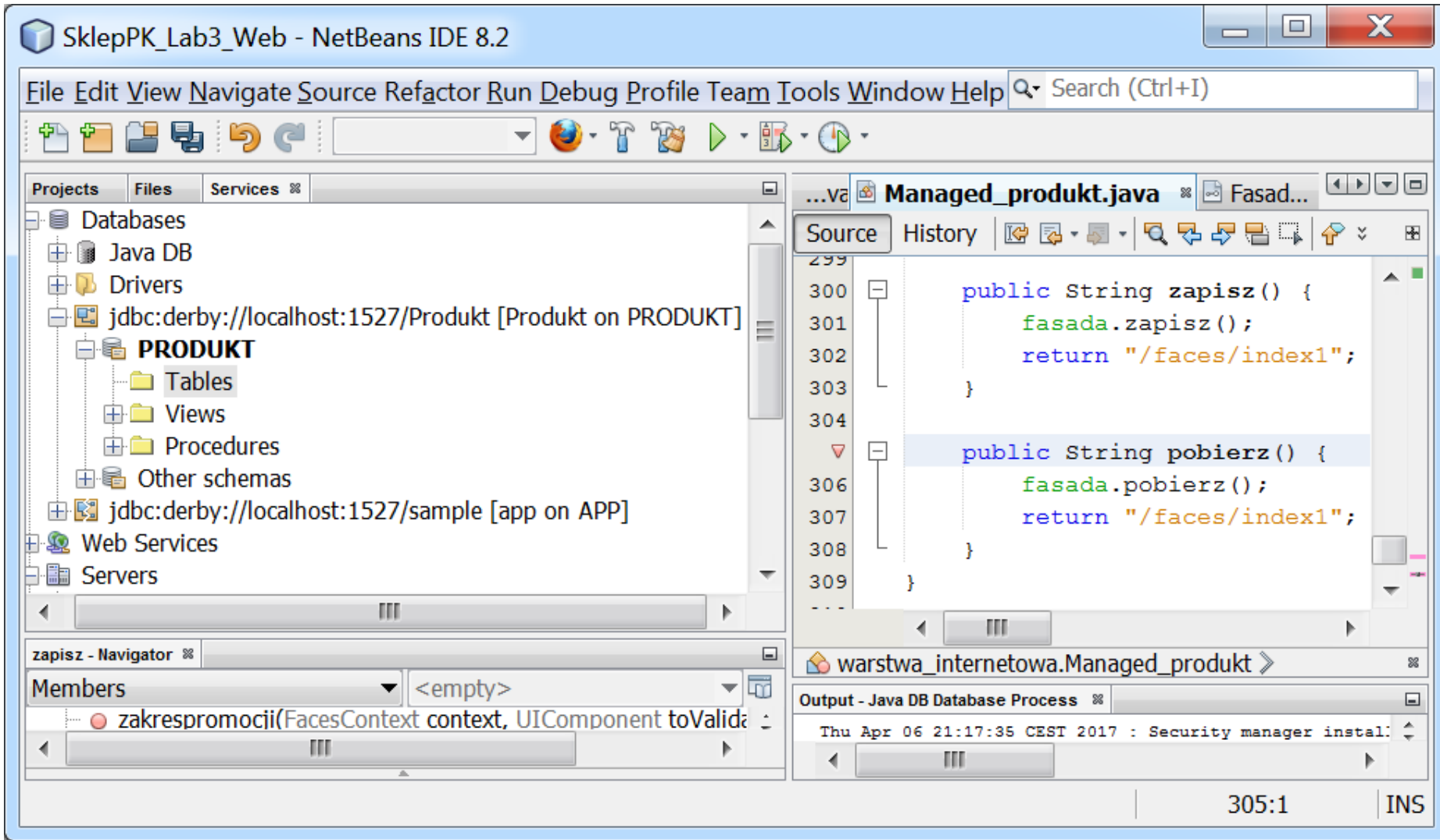
baza.zapisz=Zapisz do bazy danych

baza.pobierz=Pobierz z bazy danych

8.(8). 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>
```

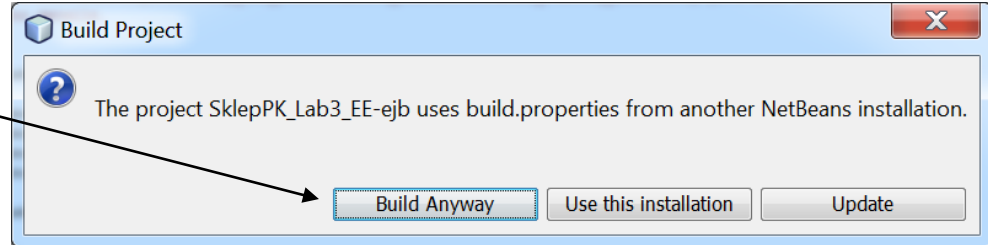
9.(1). Baza danych jest pusta przed uruchomieniem aplikacji



9. (2). 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.

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

The screenshot shows the NetBeans IDE 8.2 interface. On the left, the 'Databases' tree is expanded to show the 'PRODUKT' database. The 'PRODUKT1' table is selected, and a context menu is open over it, with 'View Data...' highlighted. The main editor window shows a SQL query: `SELECT * FROM PRODUKT.PRODUKT1 FETCH FIRST 100 ROWS ONLY;`. Below the query, a table structure is visible with columns: #, ID, CENA, DATA_PRODUKCJI, NAZWA, and PROMOCJA. The bottom status bar shows the execution output: 'Executed successfully in 0.015 s. Fetching resultset took 0 s. Line 1, column 1. Execution finished after 0.422 s, no errors occurred.'

9.(4). 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;

NetBeans IDE 8.2

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

Search (Ctrl+I)

Files Projects Services

jdbc:derby://localhost:1527/Produkt [F

PRODUKT

- Tables
 - PRODUKT1
 - ID
 - CENA
 - DATA_PRODUKCJI
 - NAZWA
 - PROMOCJA
 - Indexes
 - Foreign Keys
 - SEQUENCE**
 - SEQ_NAME
 - SEQ_COUNT
- Indexes
 - SQL170325233539710
- Foreign Keys
- Views
- Procedures

Navigator

- .right_content
- Elements

SQL 4 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/Produkt [Produkt on PR...

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

2

SELECT * FROM PRODUKT."SE...

Max. rows: 100 | Fetched Rows: 1 | Matching Rows:

#	SEQ_NAME	SEQ_COUNT
1	SEQ_GEN	50

Output

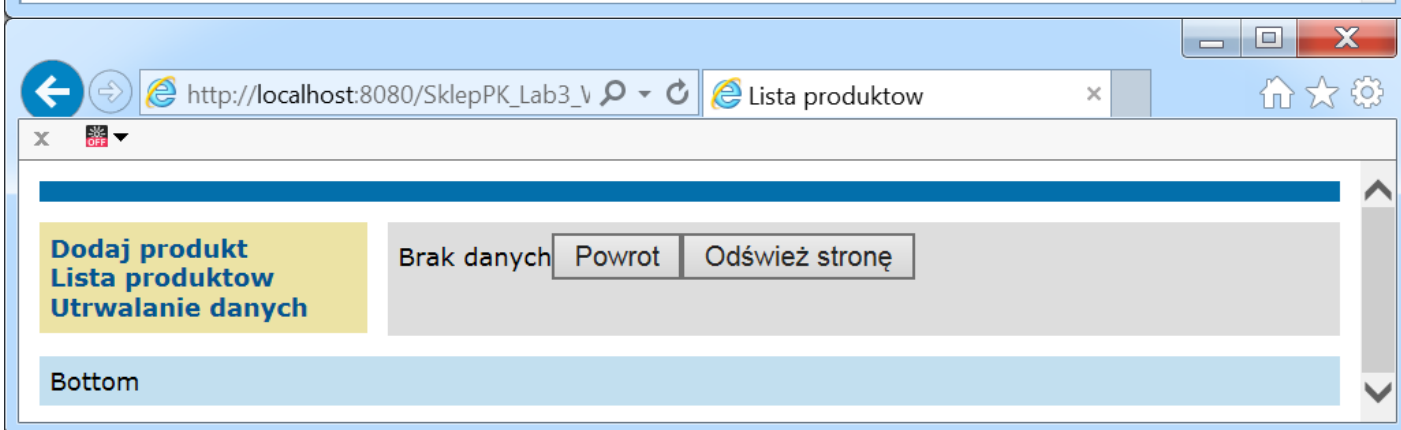
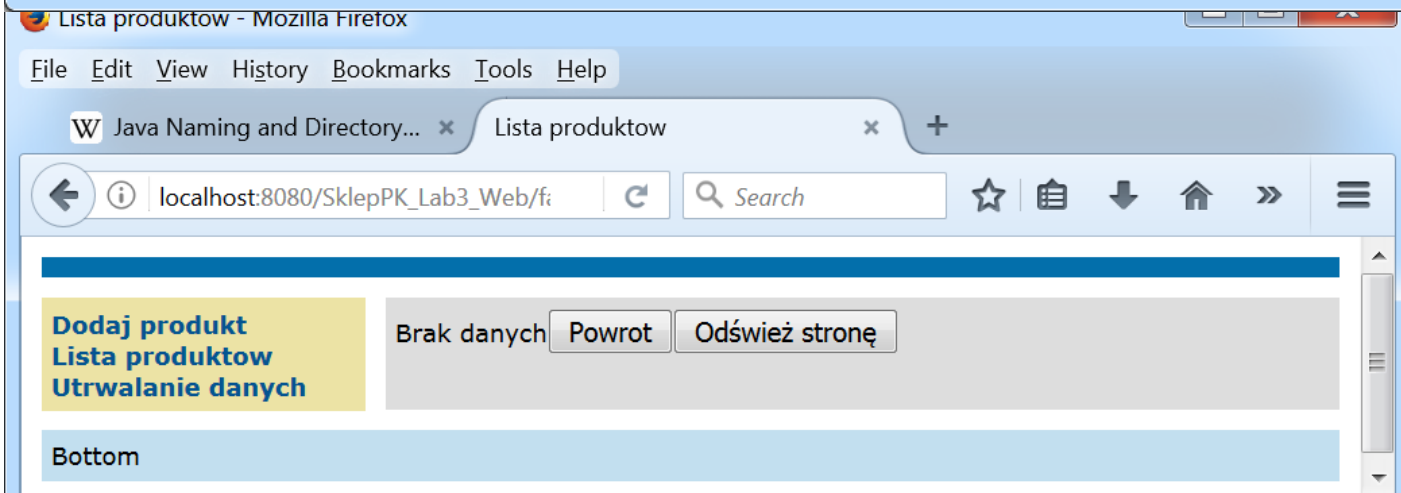
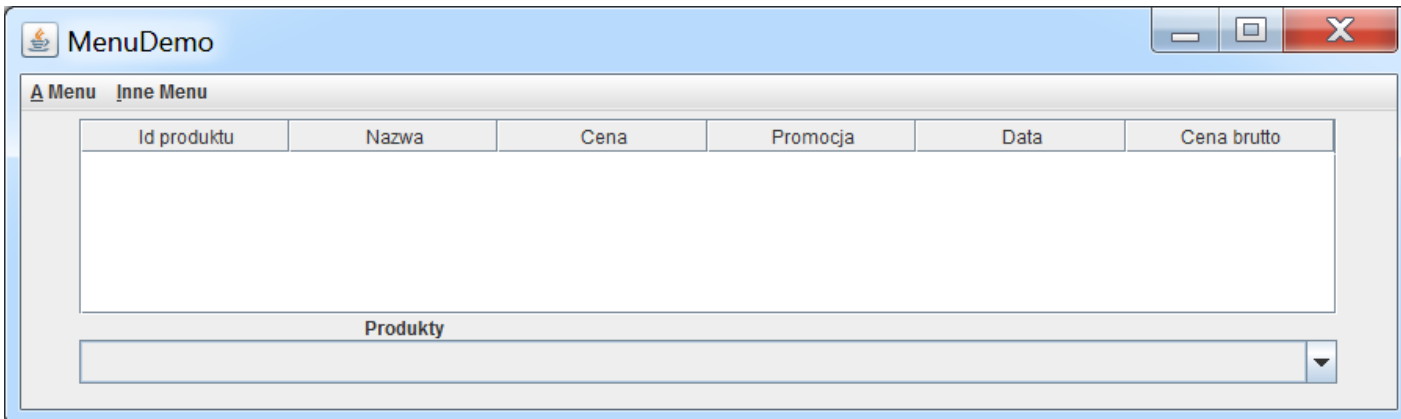
1.1 Sklep_6JPA (run) SQL 1 execution SQL 2 execution SQL 3 execution

1:1 INS

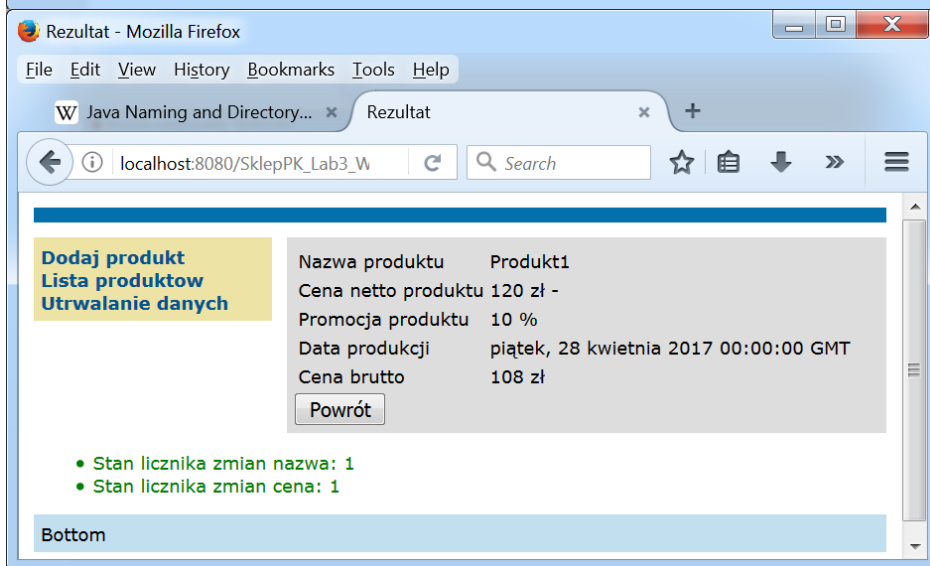
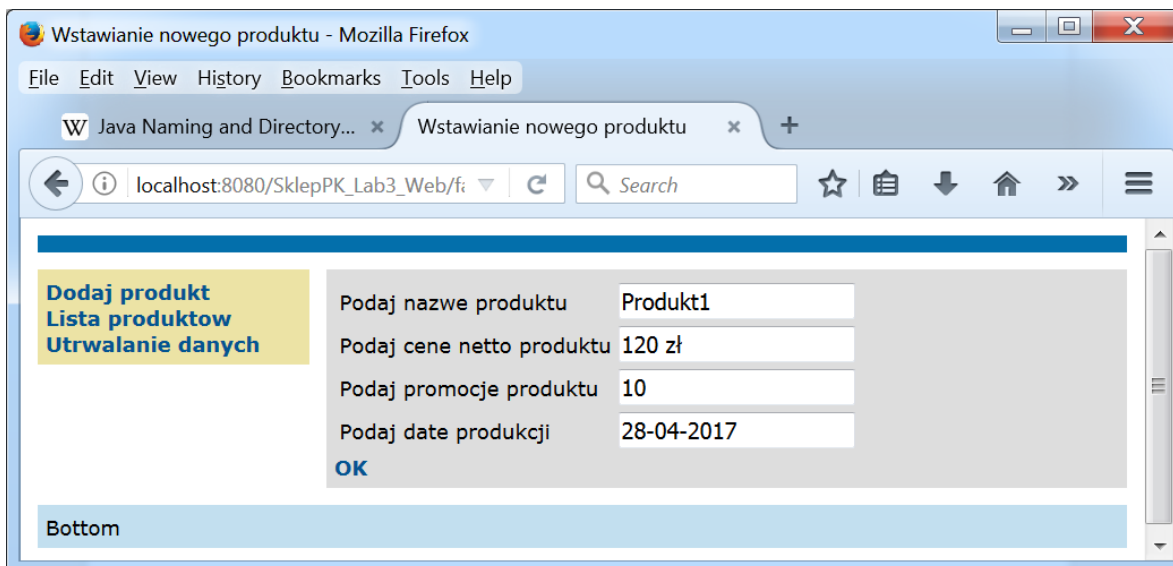
10. (1). Uruchomienie programu

The image shows two overlapping windows. The top window is titled "MenuDemo" and contains a menu bar with "Menu" and "Inne Menu" options. The bottom window is a web browser with two tabs: "Java Naming and Directory..." and "Sklep 4". The active tab "Sklep 4" shows a web page with a navigation menu on the left containing "Dodaj produkt", "Lista produktów", and "Utrwalanie danych", and a main content area labeled "Content". The browser's address bar shows "localhost:8080/SklepPK_Lab3_Web/".

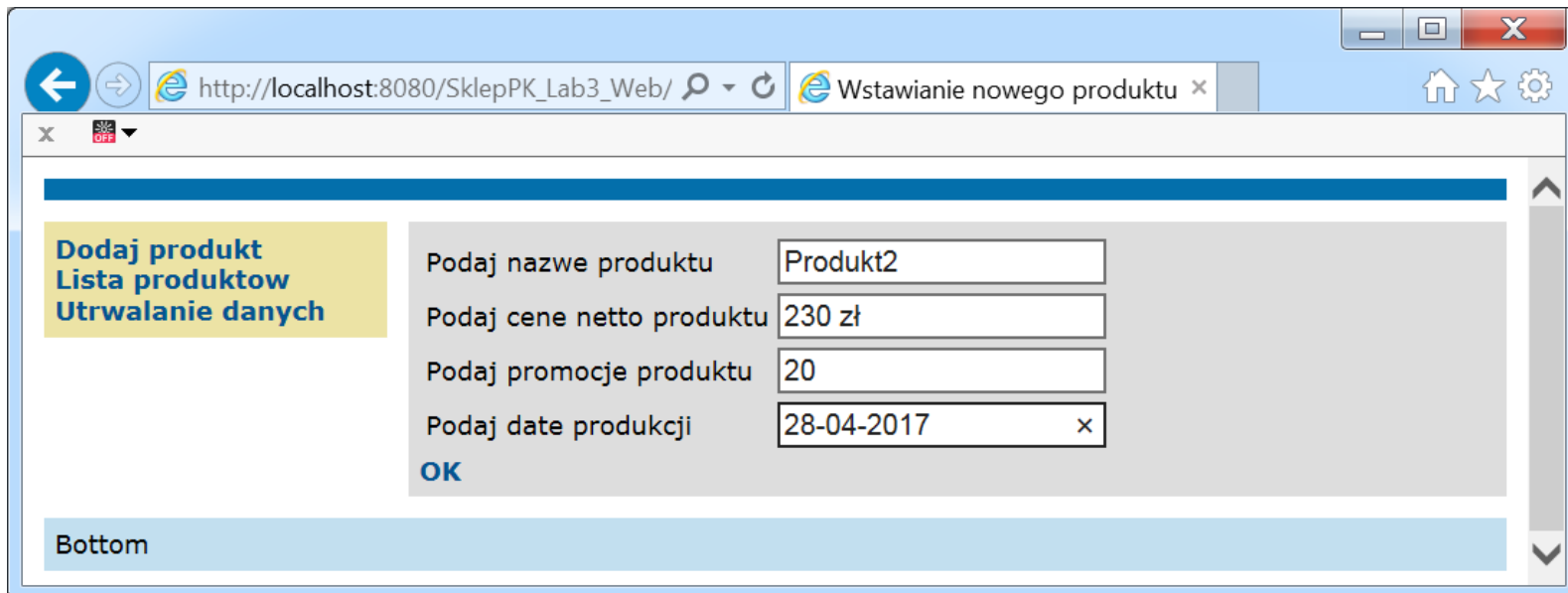
10. (2). Uruchomienie programu – brak danych



10. (3). Wprowadzenie danych w jednej z uruchomionych aplikacji



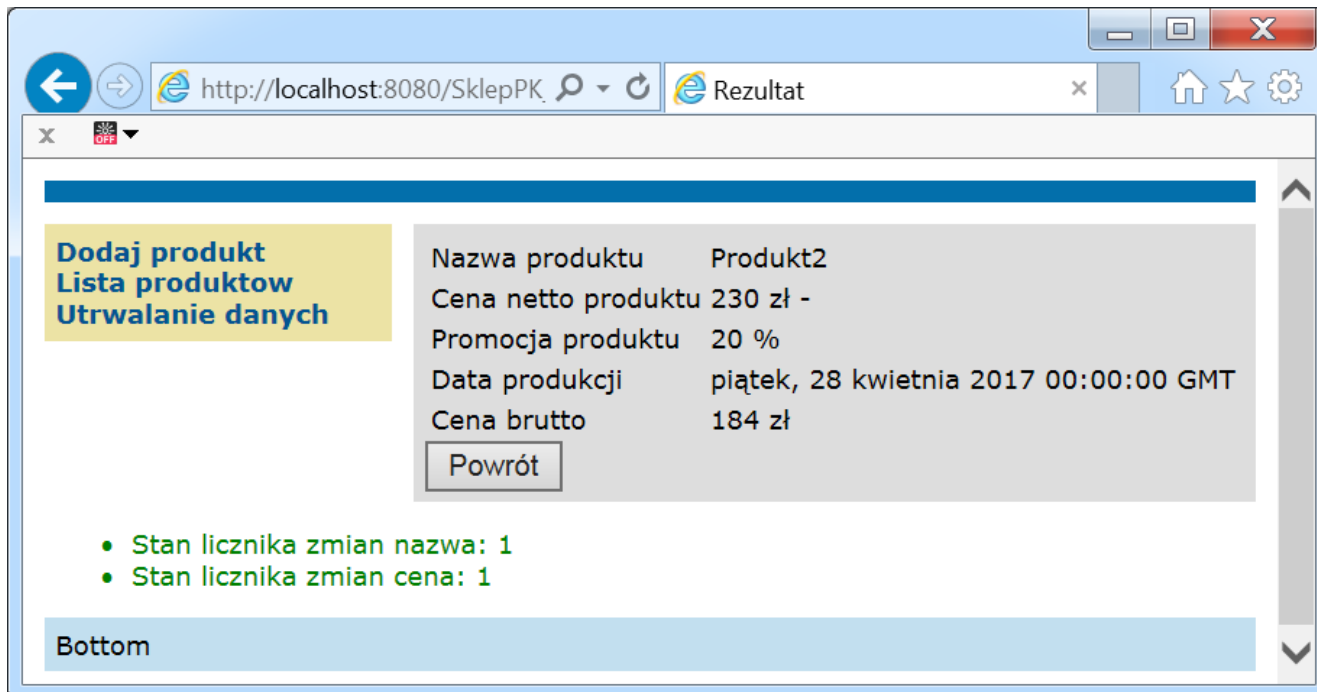
10. (4). Wprowadzenie danych w drugiej z uruchomionych aplikacji



The screenshot shows a web browser window with the address bar containing `http://localhost:8080/SklepPK_Lab3_Web/` and the page title "Wstawianie nowego produktu". The page content includes a navigation menu on the left with items: "Dodaj produkt", "Lista produktów", and "Utrwalanie danych". The main form area contains the following fields and values:

Podaj nazwe produktu	<input type="text" value="Produkt2"/>
Podaj cene netto produktu	<input type="text" value="230 zł"/>
Podaj promocje produktu	<input type="text" value="20"/>
Podaj date produkcji	<input type="text" value="28-04-2017"/>

Below the form is an "OK" button. At the bottom of the page, there is a "Bottom" label.

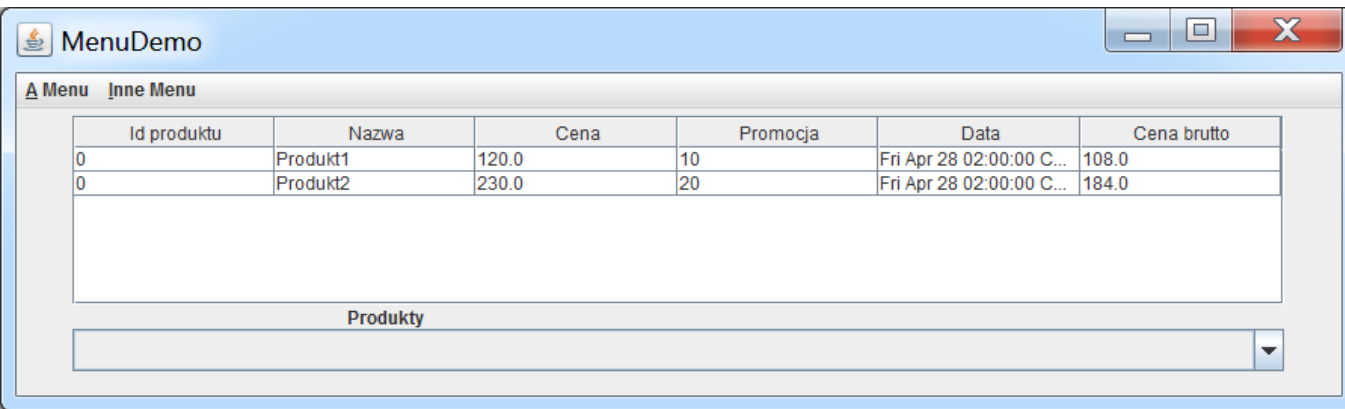


The screenshot shows a web browser window with the address bar containing `http://localhost:8080/SklepPK` and the page title "Rezultat". The page content includes the same navigation menu as the previous screenshot. The main area displays the following product details:

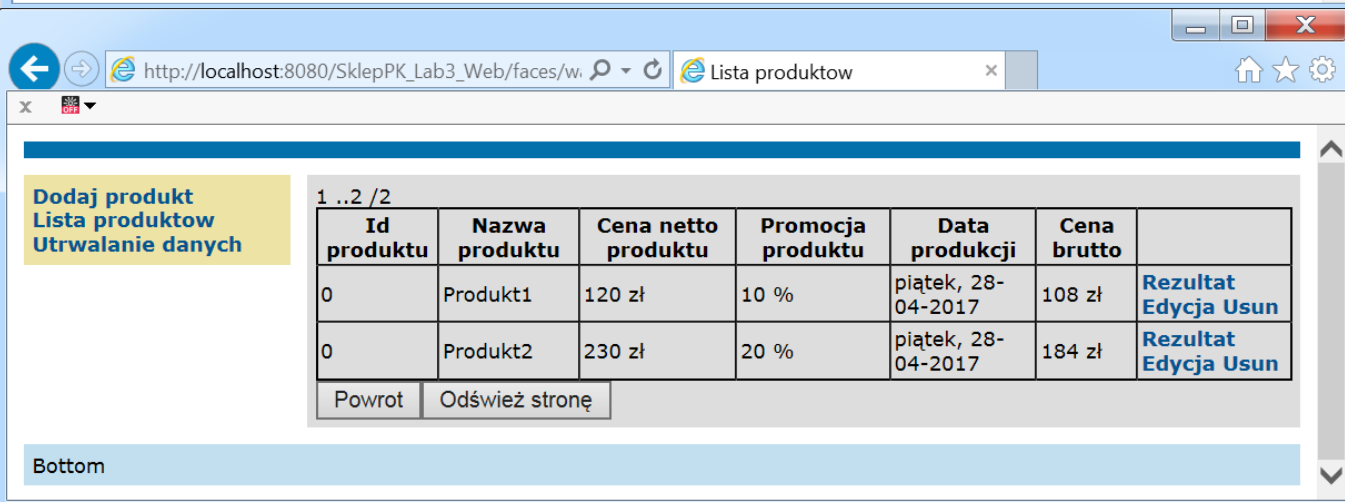
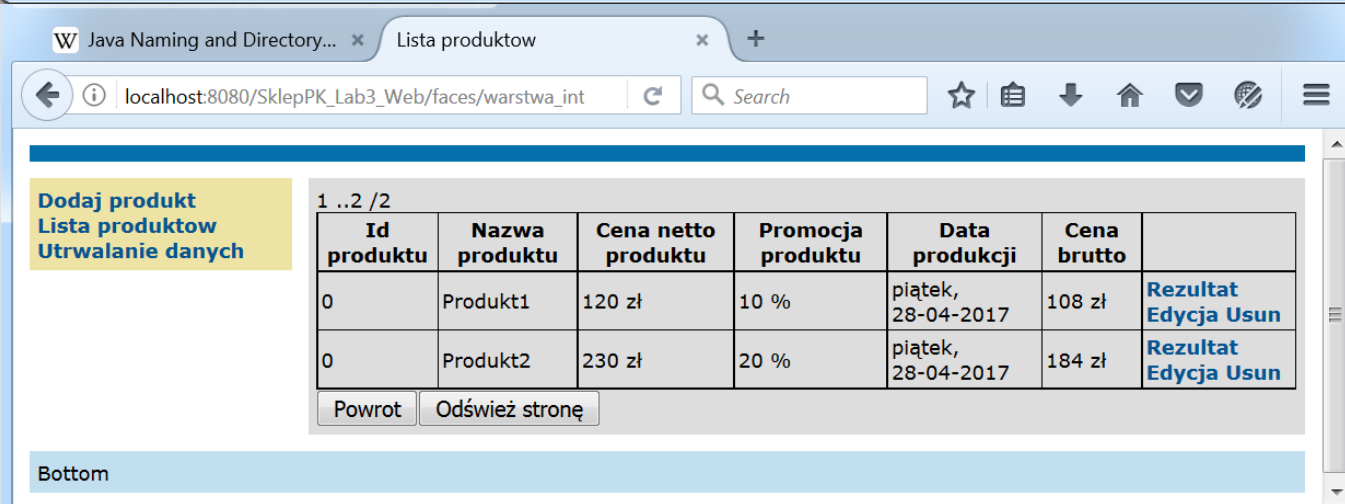
Nazwa produktu	Produkt2
Cena netto produktu	230 zł -
Promocja produktu	20 %
Data produkcji	piątek, 28 kwietnia 2017 00:00:00 GMT
Cena brutto	184 zł

Below the details is a "Powrót" button. At the bottom of the page, there is a "Bottom" label and a list of status messages:

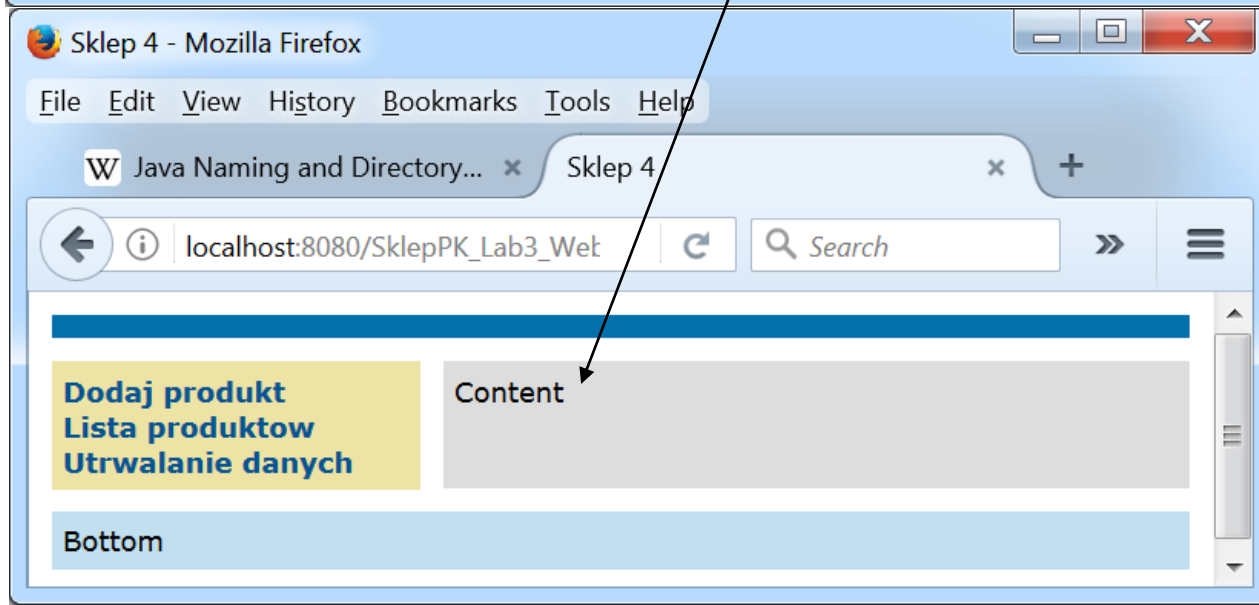
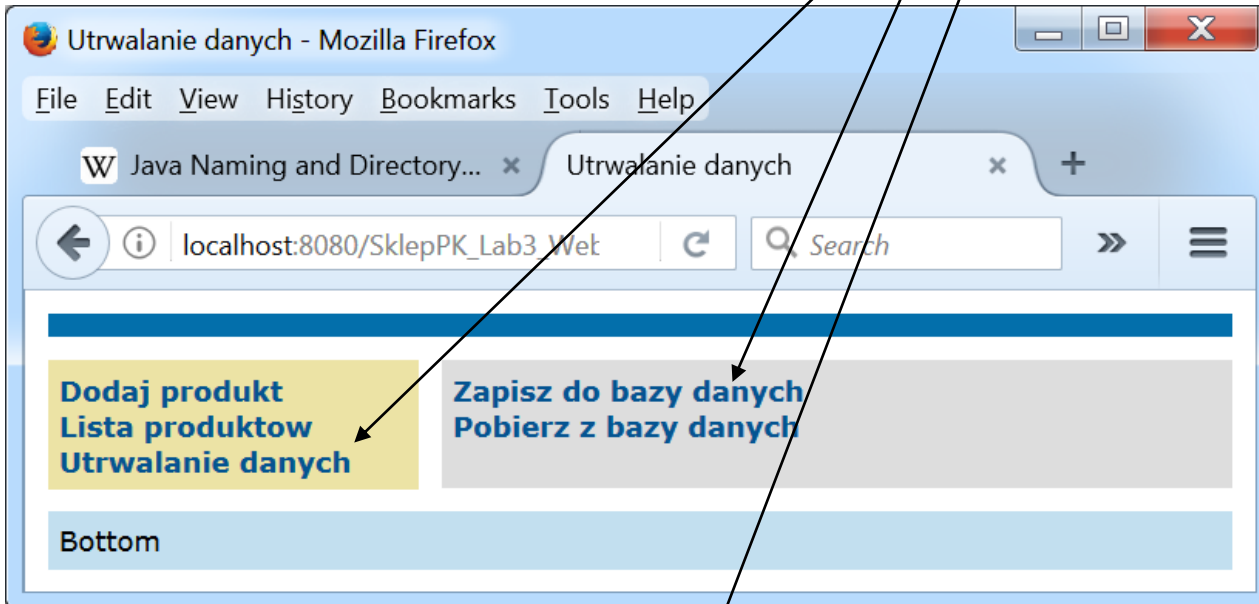
- Stan licznika zmian nazwa: 1
- Stan licznika zmian cena: 1



10. (5). Dane widoczne w każdej z uruchomionych aplikacji klienckich



10.(6). Zapis do bazy danych



10.(6). 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 displays a tree view of a database named 'PRODUKT'. Under 'Tables', the 'PRODUKT1' table is selected. The main editor window shows a SQL query: `SELECT * FROM PRODUKT.PRODUKT1 FETCH FIRST 100 ROWS ONLY;`. Below the query, the results are displayed in a table with the following data:

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

Below the table, the 'Output' pane shows the execution log for 'Sklep_GUIPK_lab3_EE_Desktop (run)'. The log indicates that the query was executed successfully in 0 seconds, with no errors.

Sklep_GUIPK_lab3_EE_Desktop (run) running... 1:1 INS

10.(7). Dodanie kolejnej danej w aplikacji desktopowej

The image shows two windows from a desktop application. The top window, titled 'MenuDemo', contains a form for adding a new product. The fields are: Nazwa (Produkt3), Cena (345), Promocja (30), and Data (28-04-2017). A 'Dodaj produkt' button is at the bottom. The bottom window is a Mozilla Firefox browser displaying the 'Lista produktow' page. It features a table with product details and navigation buttons.

MenuDemo Form Data:

Nazwa	Produkt3
Cena	345
Promocja	30
Data	28-04-2017

Lista produktow Table:

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

Web Browser Table:

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

10.(8). Wyświetlenie danych przechowywanych w aplikacji oraz po pobraniu z bazy danych

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

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

http://localhost:808 Utrwalanie danych

Dodaj produkt
Lista produktow
Utrwalanie danych

Zapisz do bazy danych
Pobierz z bazy danych

Bottom

http://localhost:808 Sklep 4

Dodaj produkt
Lista produktow
Utrwalanie danych

Content

Bottom

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

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

10.(9). 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

Dodaj produkt
Lista produktow
Utrwalanie danych

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