

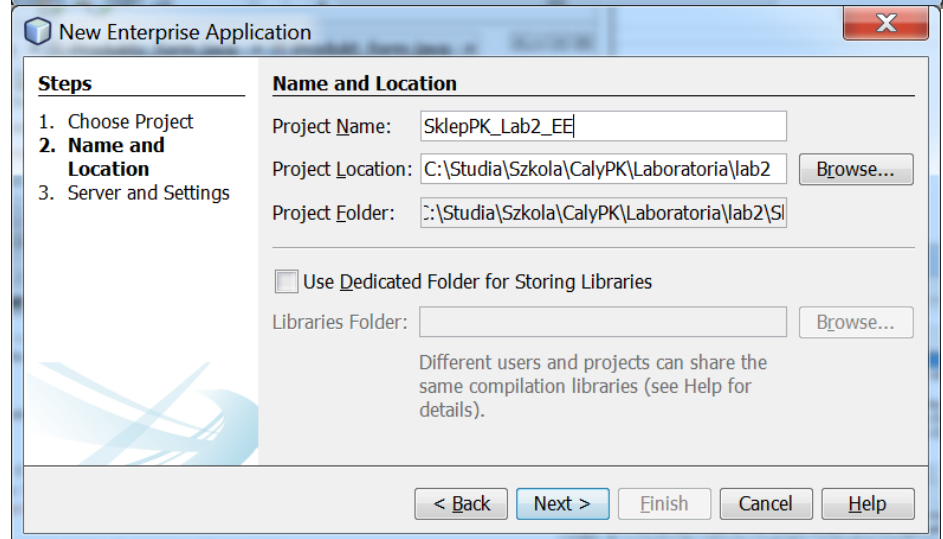
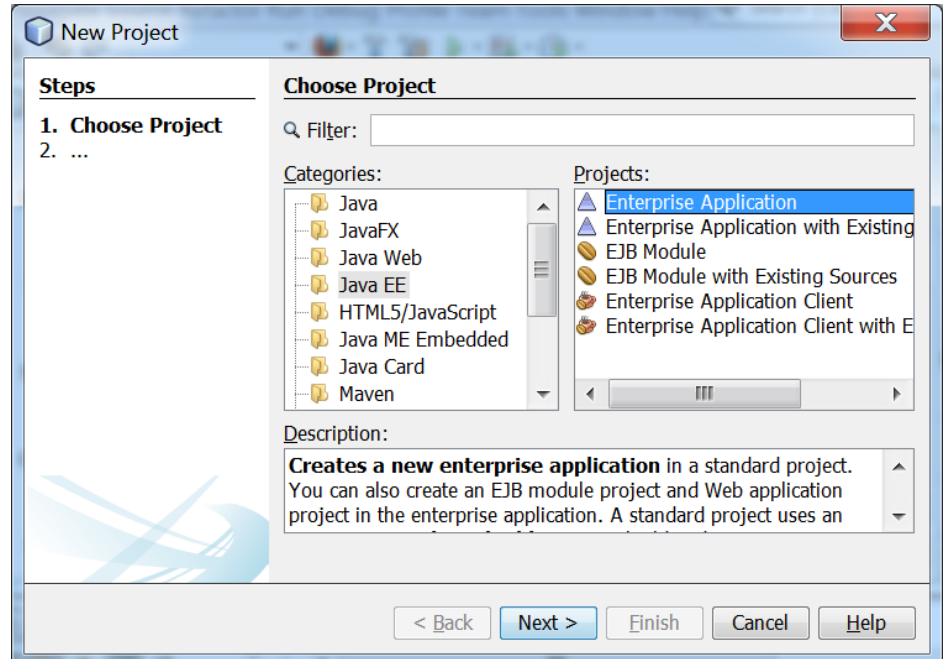
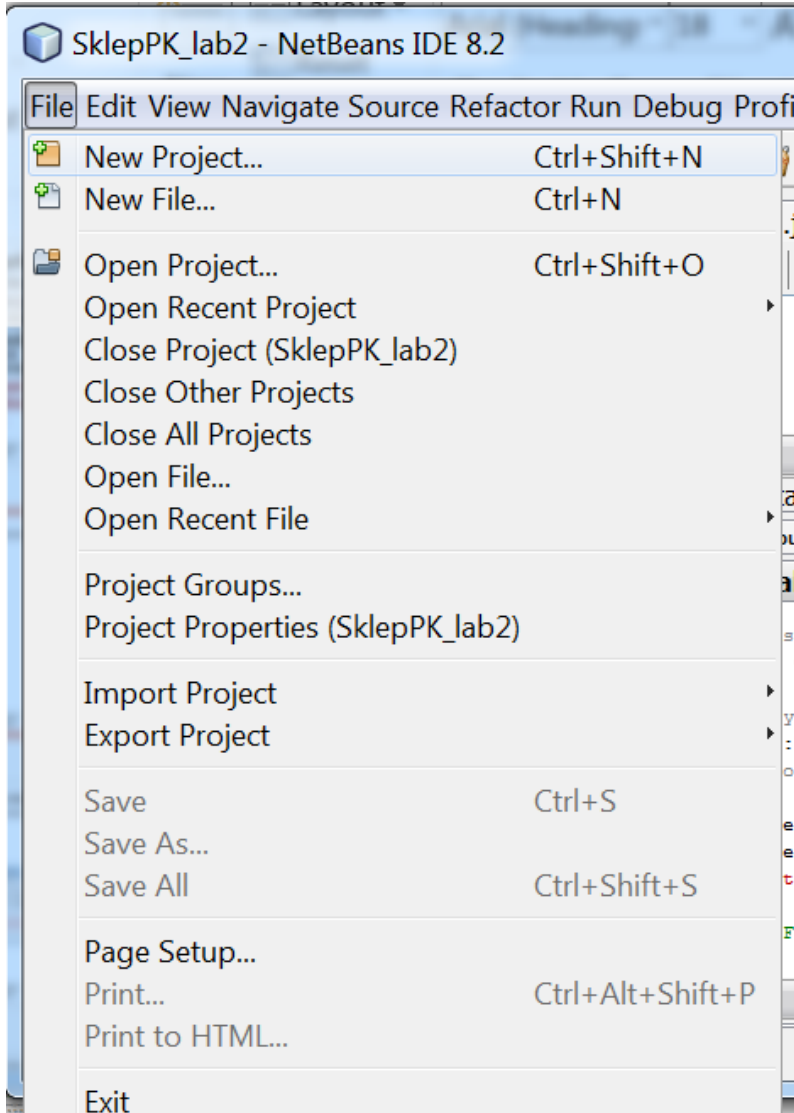
Wykonanie aplikacji wielowarstwowej na platformie Java EE z warstwą klienta zawierającą klienta internetowego i desktopowego na podstawie

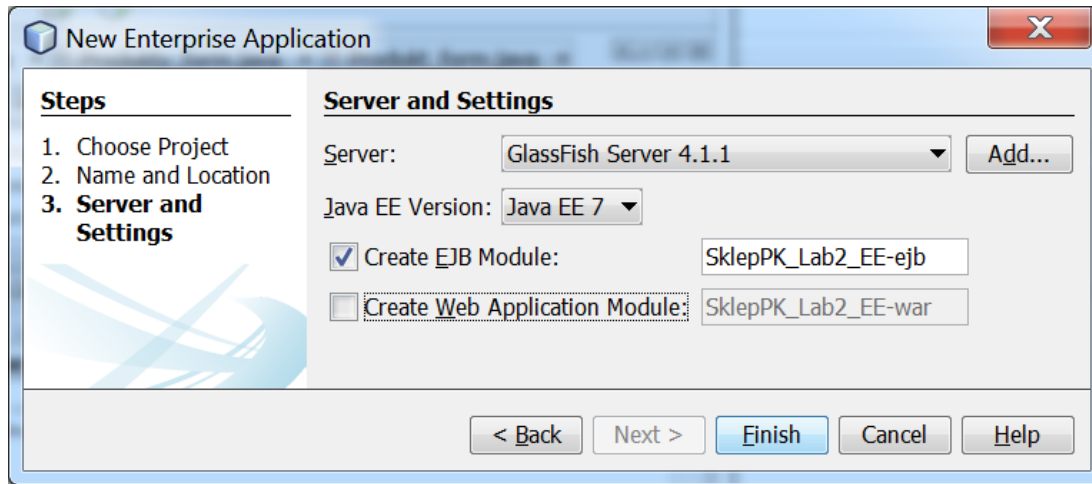
<https://docs.oracle.com/javase/7/JEETT.pdf>

Programowanie komponentowe Lab2

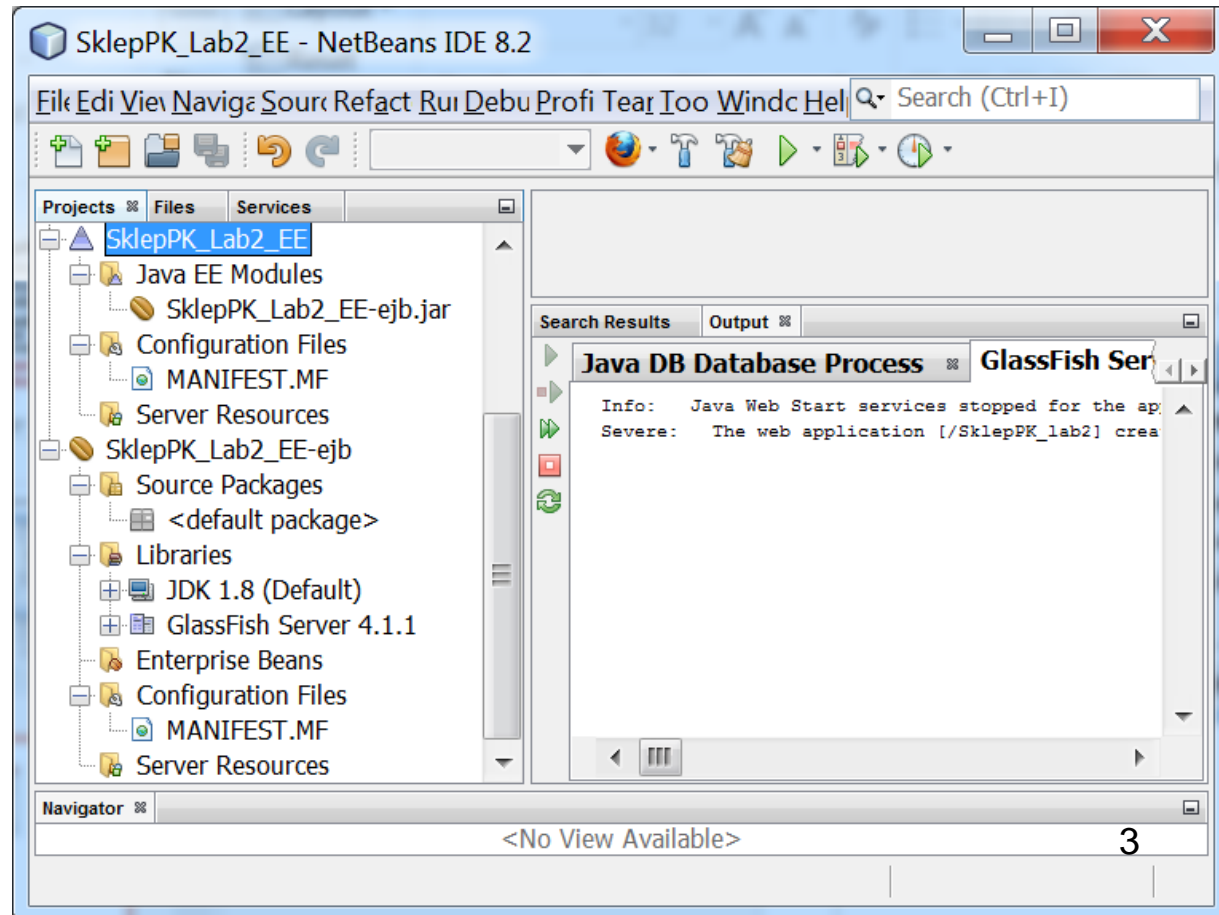
1. Wykonanie projektu typu Enterprise Application.

1.1. Należy wybrać: **File/New Project.../**. Następnie w Categories: **Java EE** i w Projects: **Enterprise Application**. Następnie, należy podać nazwę projektu w polu **Project Name (SklepPK_Lab2_EE)** i położenie projektu w polu **Project Location**.

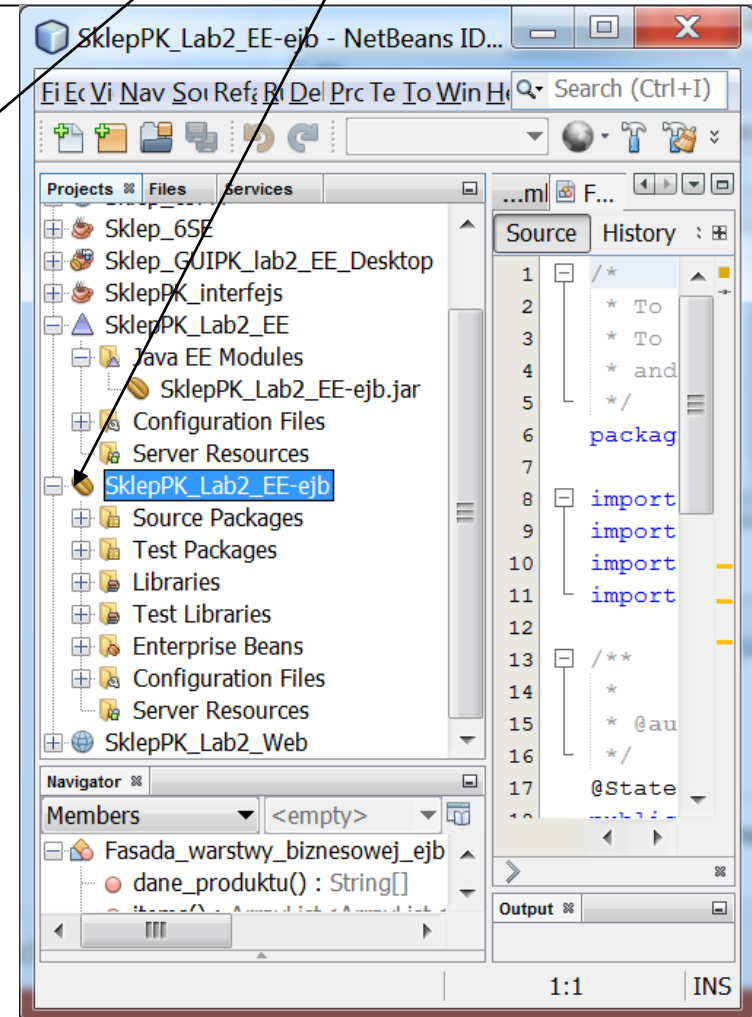
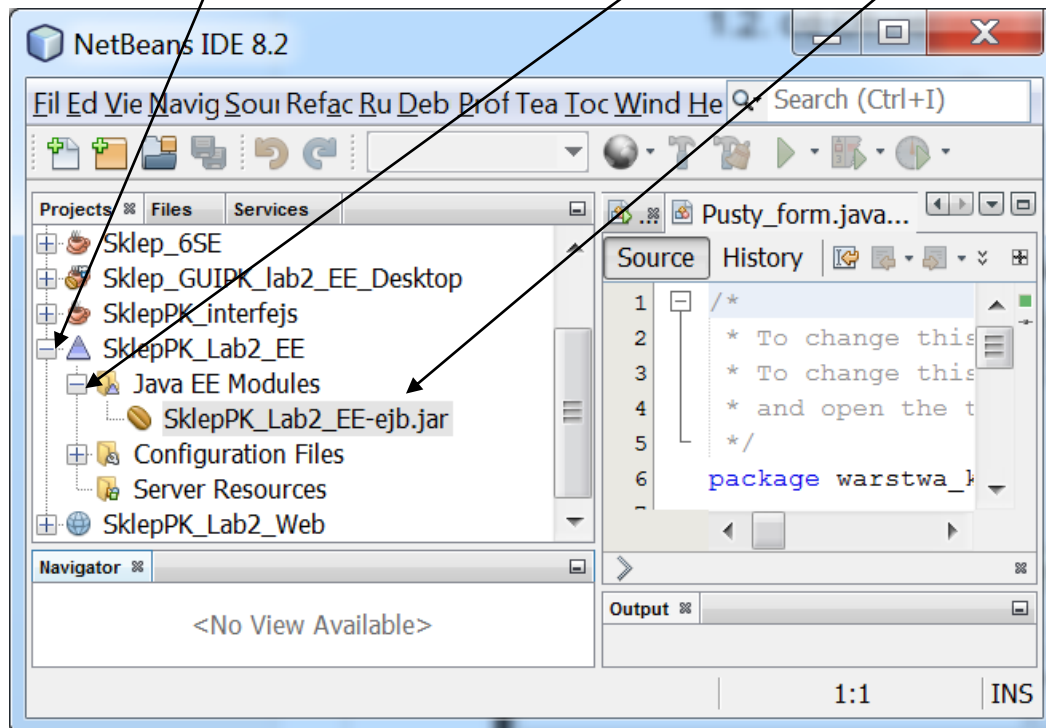




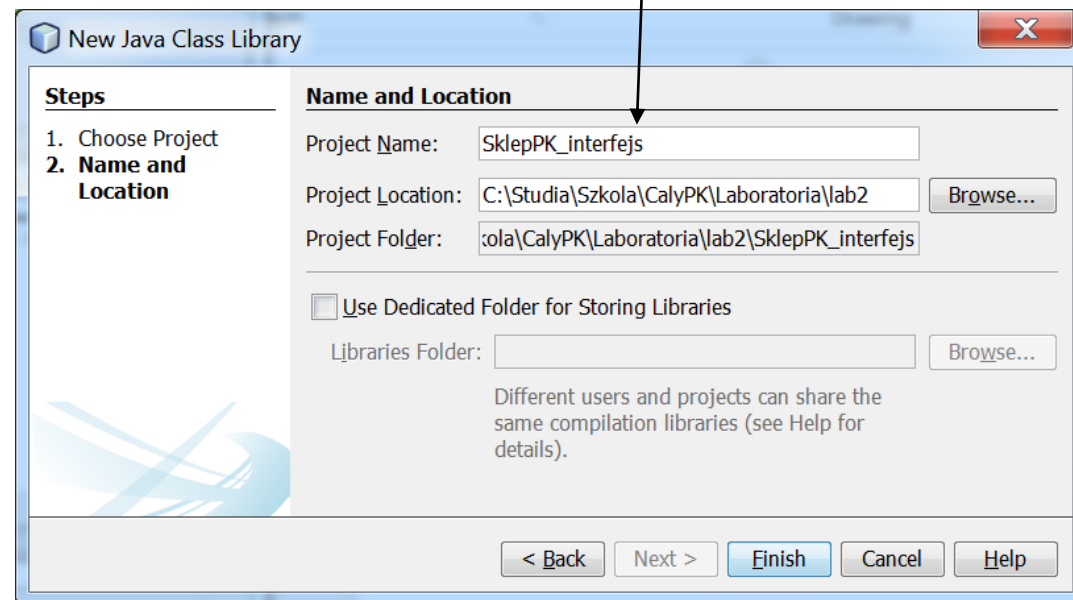
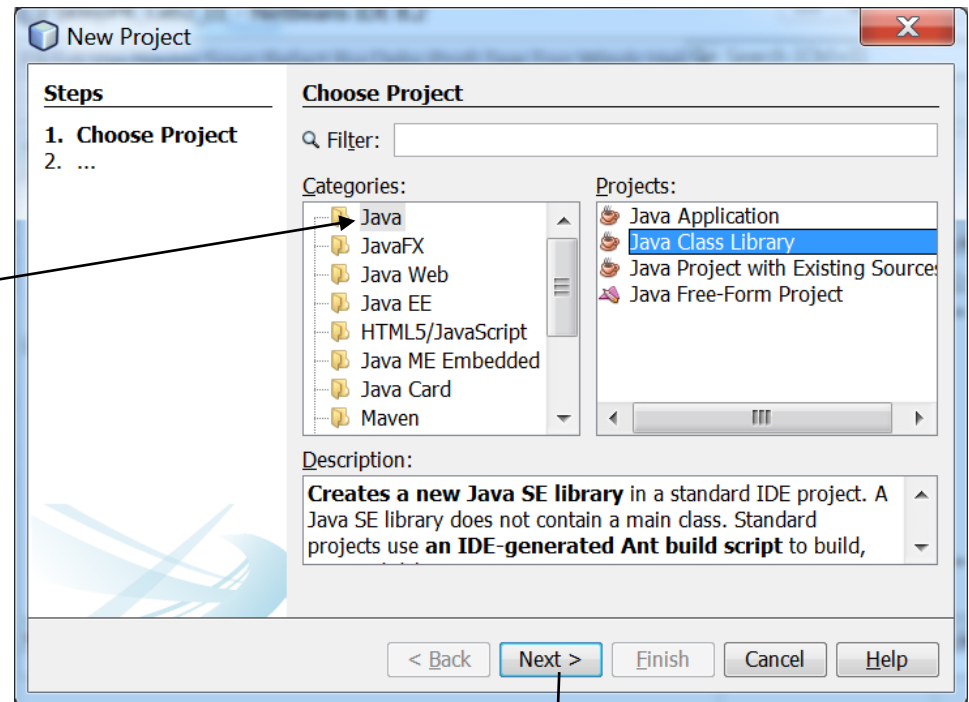
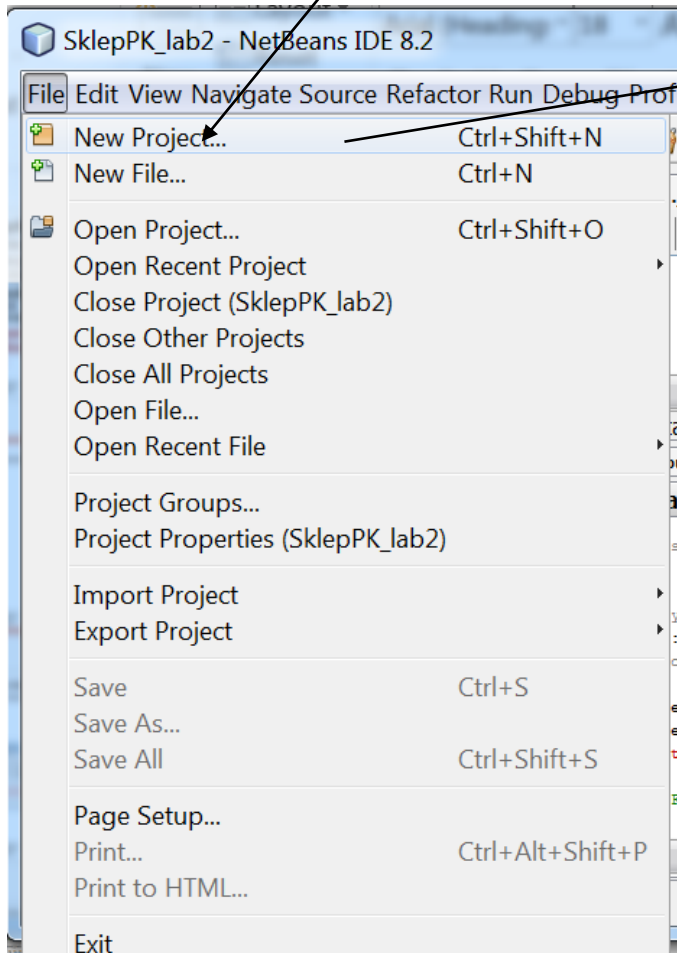
1.2. Należy wykonać jedynie moduł EJB, w celu definiowania w nim komponentów EJB rodzaju **Session** (zaznaczenie **Create EJB Module**).



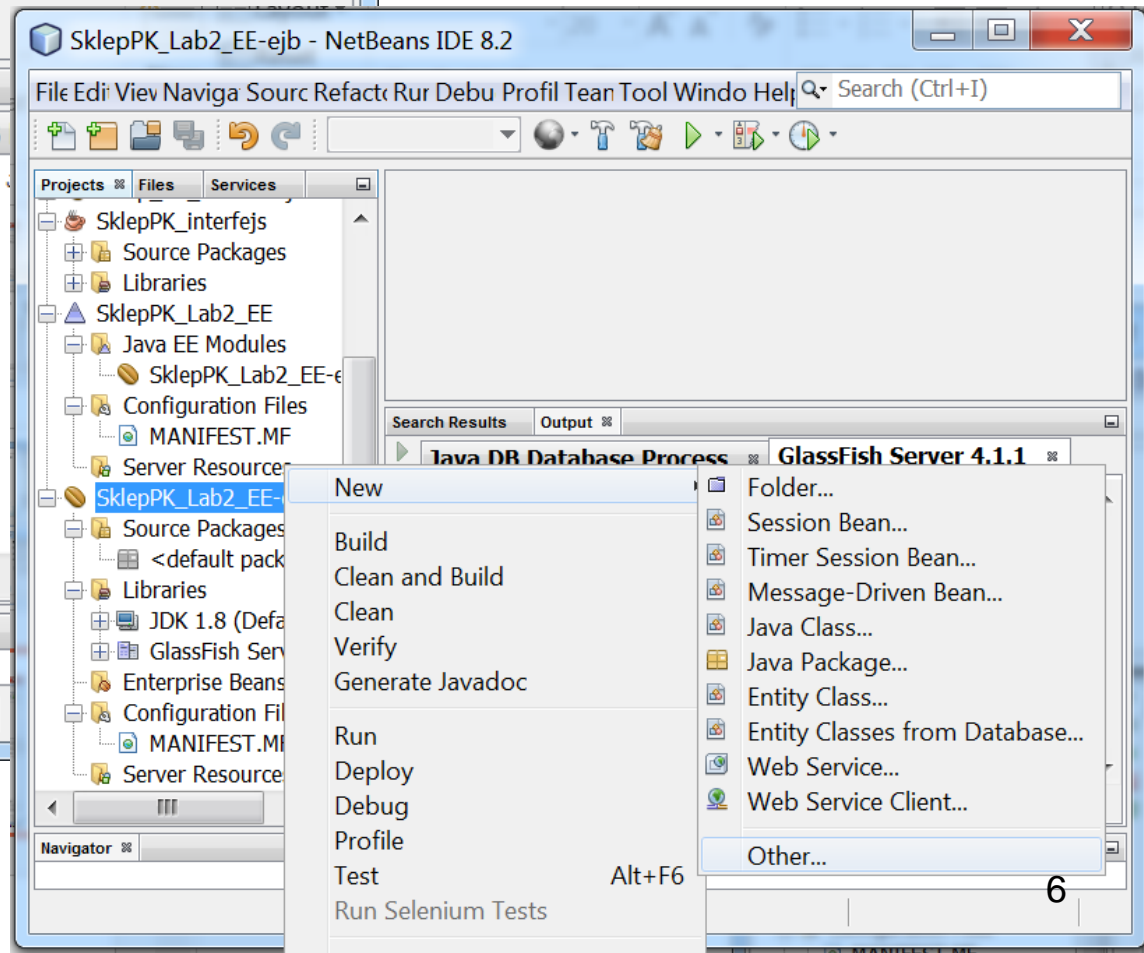
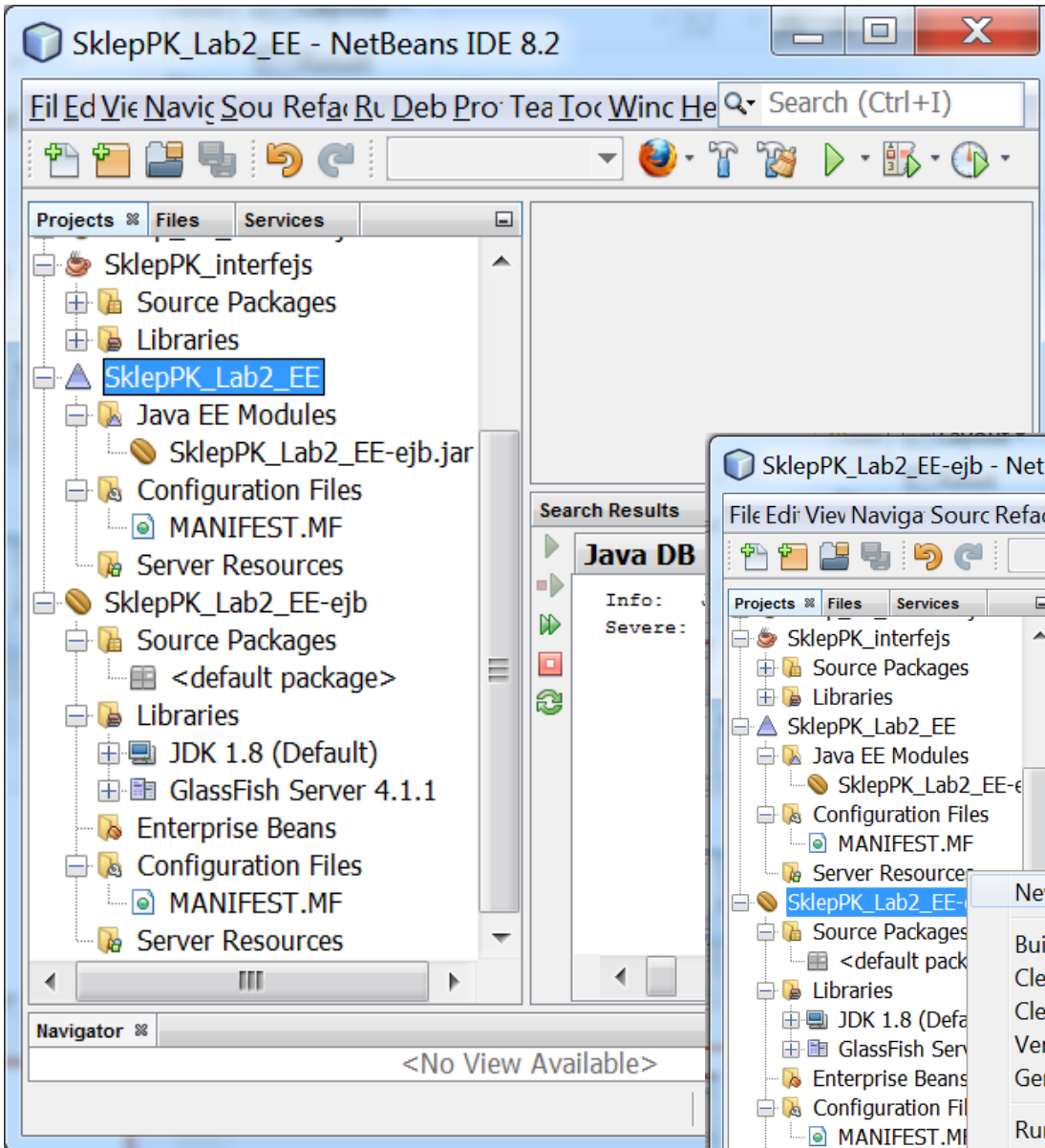
1.2. cd Otwieranie modułu EJB w oknie zakładki **Projects** – należy rozwinąć zawartość projektu **SklepPK_Lab2_EE**, następnie zawartość folderu **Java EE Modules** i dwukrotnie kliknąć na pozycję **SklepPK_Lab2_EE-ejb**. W oknie zakładki **Projects** pojawi się pozycja modułu EJB - **SklepPK_Lab2_EE-ejb**.

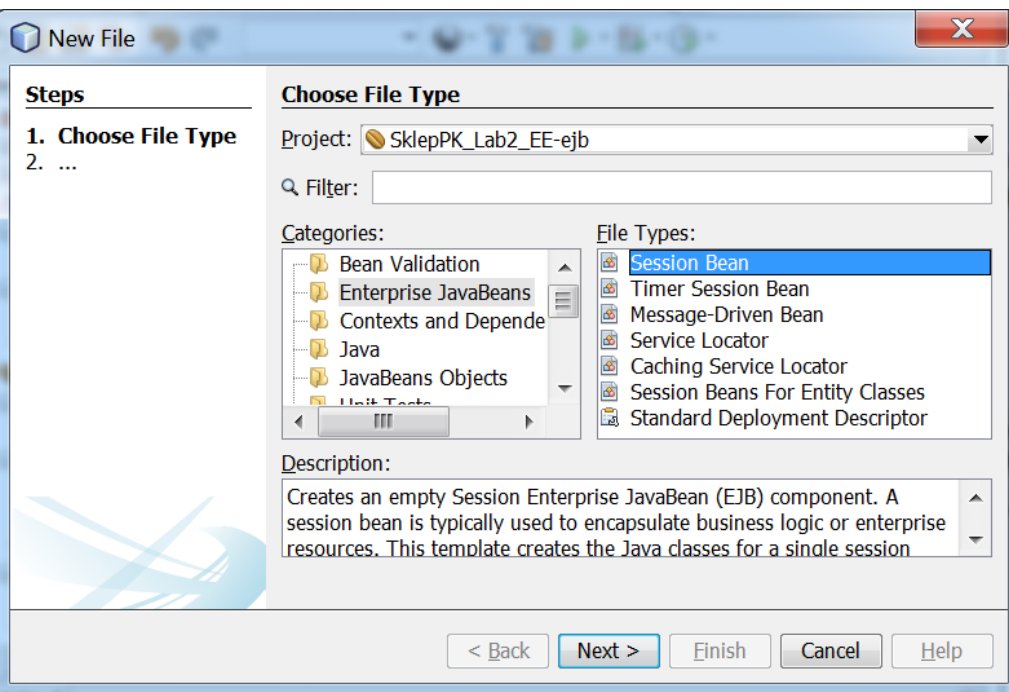


1.3. Wykonanie projektu Java SE typu Java Class Library (**SklepPK_interfejs**) w celu zdefiniowania w nim w kolejnych krokach interfejsu komponentu EJB.



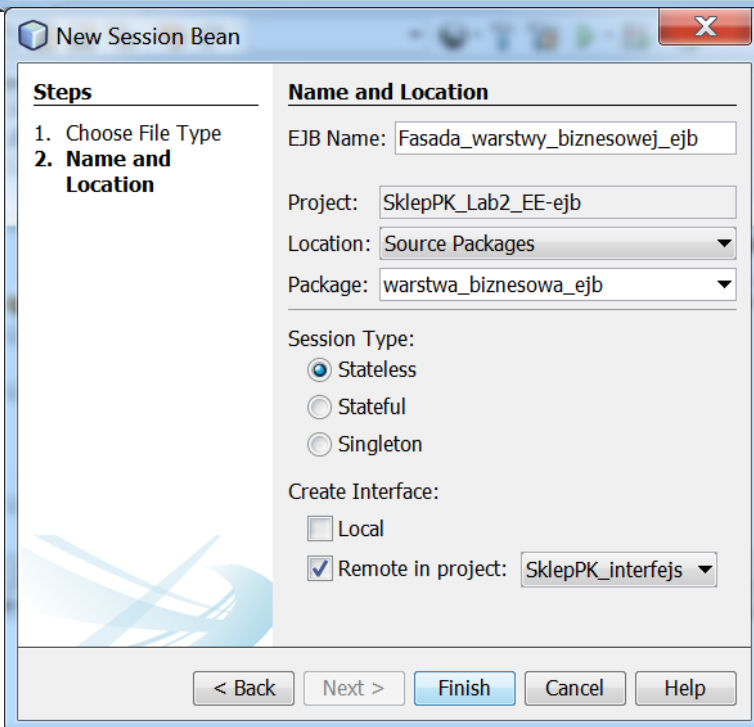
1.4. Definicja komponentu EJB typu Session w projekcie typu EJB (wybranie pozycji **New/Other**)



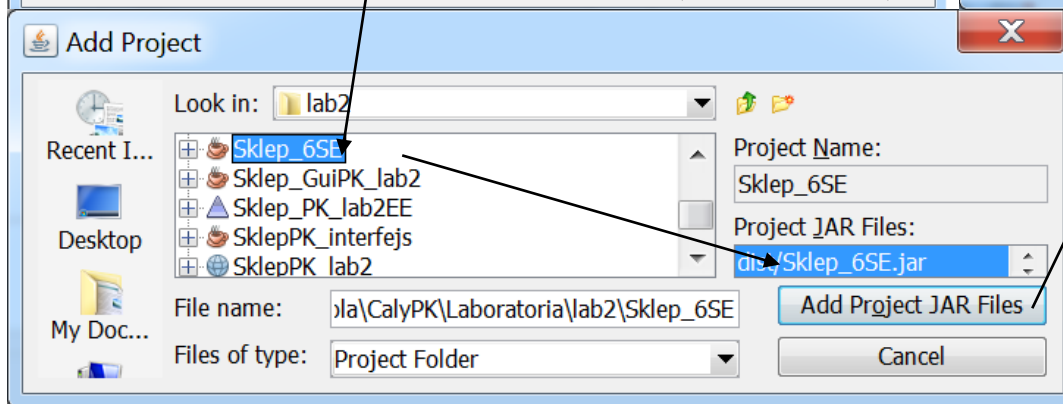
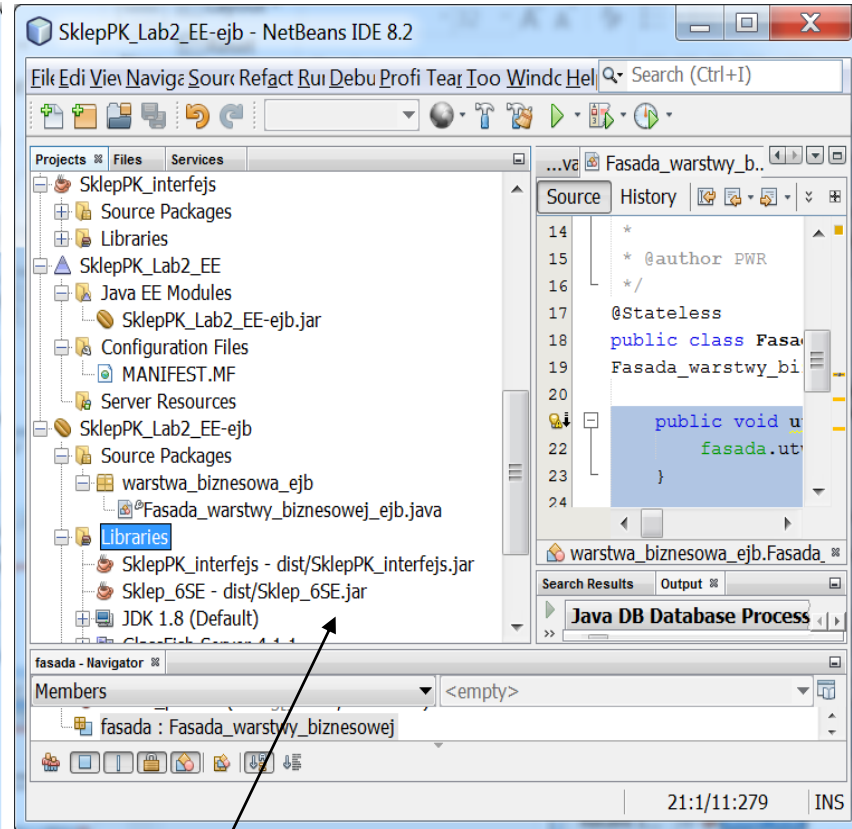
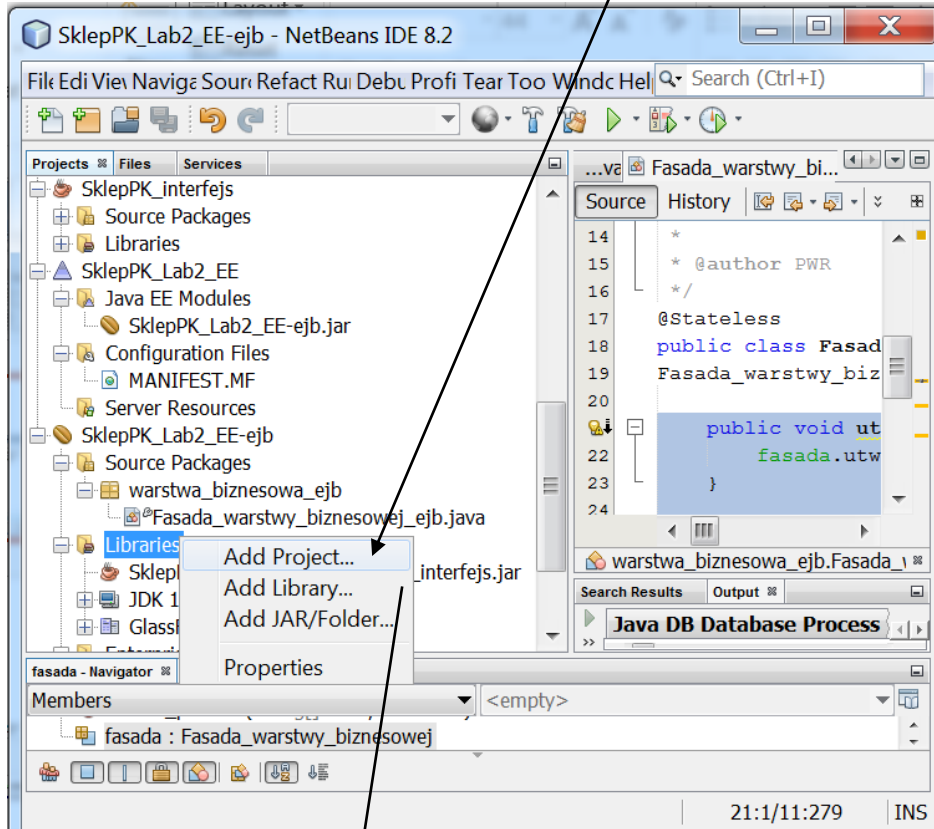


1.5. Należy wybrać w **Categories** pozycję **Enterprise JavaBeans**, a w **File Types** pozycję **Session Bean** i kliknąć na klawisz **Next**. W polu **EJB Name** kolejnego formularza wprowadzić nazwę komponentu:

Fasada_warstwy_biznesowej_ejb podając nazwę pakietu w polu **Package**: ***warstwa_biznesowa_ejb***. Należy wybrać **Session Type: Stateless** i interfejs w **Create Interface: Remote** oraz projekt z listy projektów, w którym zostanie zdefiniowany interfejs komponentu Sesion o zdalnym dostępie: **SklepPK_interfejs (p.1.3)**.



1.6. W projekcie typu EJB należy dodać jako bibliotekę, kod projektu **Sklep_6SE z lab1**, zawierający kod logiki biznesowej (**Libraries/Add Project..** i następnie w kolejnym formularzu z listy należy wybrać projekt **Sklep_6SE** i kliknąć na przycisk **Add Project JAR Files**)



1.7. W utworzonej klasie komponentu EJB typu Session o nazwie **Fasada_warstwy_biznesowej_ejb** należy wstawić kod (podobnie jak w lab1. w komponentach EJB należących do projektów **Sklep_6_1**, **Sklep_6_2**, **Sklep_6_3** z lab1)

The screenshot displays the NetBeans IDE 8.2 environment. The main editor window shows the source code for the `Fasada_warstwy_biznesowej_ejb.java` class. The code is as follows:

```
1 package warstwa_biznesowa_ejb;
2
3 import java.util.ArrayList;
4 import java.util.Date;
5 import javax.ejb.Stateless;
6 import warstwa_biznesowa.Fasada_warstwy_biznesowej;
7
8 @Stateless
9 public class Fasada_warstwy_biznesowej_ejb implements
10     Fasada_warstwy_biznesowej_ejbRemote {
11     Fasada_warstwy_biznesowej fasada = new Fasada_warstwy_biznesowej();
12
13     public void utworz_produkt(String dane[], Date data) {
14         fasada.utworz_produkt(dane, data);
15     }
16
17     public String[] dane_produktu() {
18         return fasada.dane_produktu();
19     }
20
21     public ArrayList<ArrayList<String>> items() {
22         return fasada.items();
23     }
24 }
```

The IDE interface includes a Project Explorer on the left showing the project structure, a Search Results and Output window at the bottom, and a Navigator window showing the `dane_produktu() : String[]` method.

1.7. cd Należy zmodyfikować kod metod komponentu EJB **Fasada_warstwy_biznesowej_ejb**, który teraz może być zdefiniowany za pomocą metod klasy **Fasada_warstwy_biznesowej**.

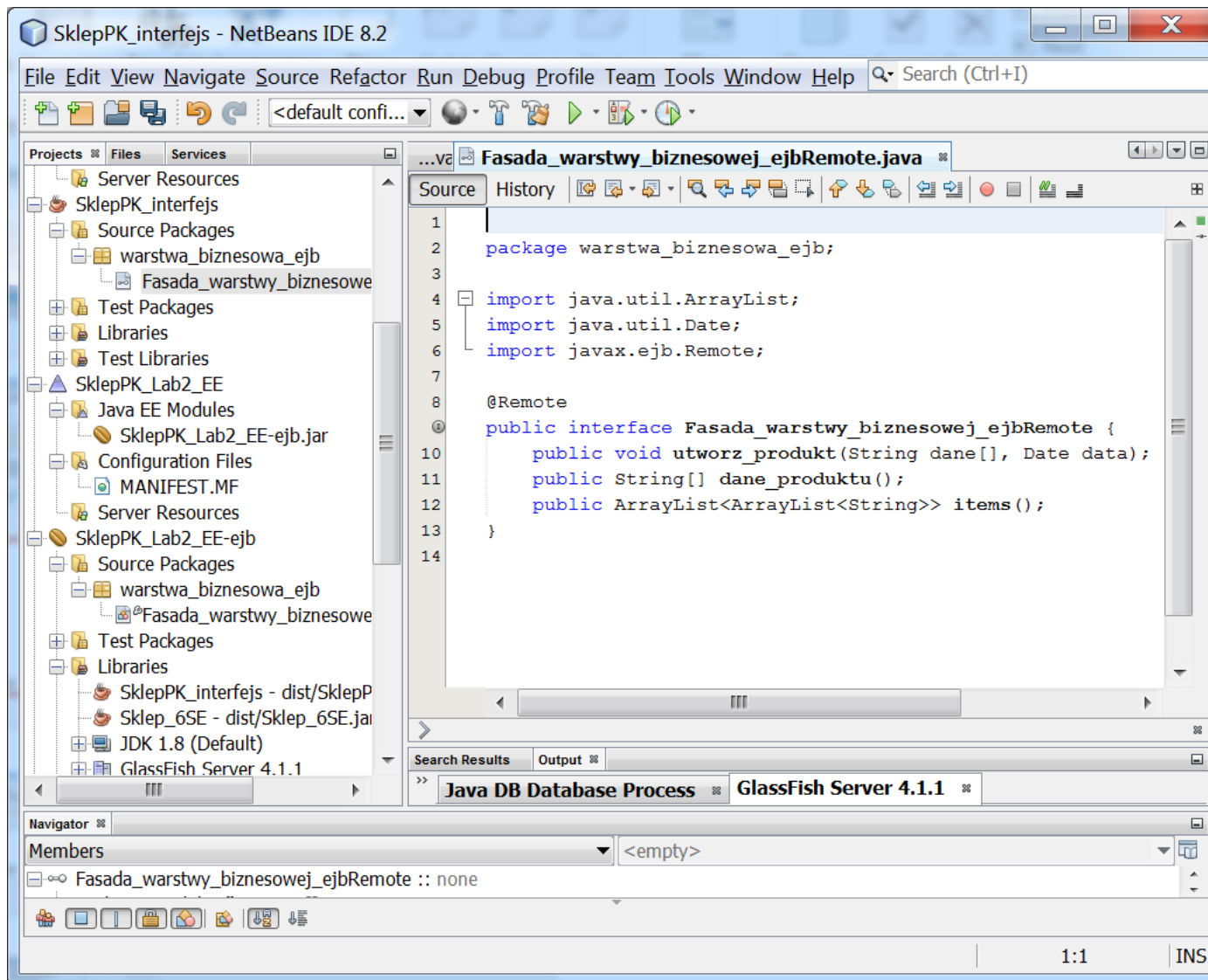
```
package warstwa_biznesowa;

import java.util.ArrayList;
import java.util.Date;
import javax.ejb.Stateless;
import warstwa_biznesowa.Fasada_warstwy_biznesowej;

@Stateless
public class Fasada_warstwy_biznesowej_ejb implements Fasada_warstwy_biznesowej_ejbRemote
{
    Fasada_warstwy_biznesowej fasada=new Fasada_warstwy_biznesowej();

    public void utworz_produkt(String dane[], Date data) {
        fasada.utworz_produkt(dane, data);
    }
    public String[] dane_produktu() {
        return fasada.dane_produktu();
    }
    public ArrayList<ArrayList<String>> items() {
        return fasada.items();
    }
}
```

1.8. Należy zdefiniować kod interfejsu **Fasada_warstwy_biznesowej_ejbRemote** metod komponentu typu **Session**, w projekcie typu **Java Class Library**, wybranym podczas definiowania tego komponentu Session (p. 1.5).



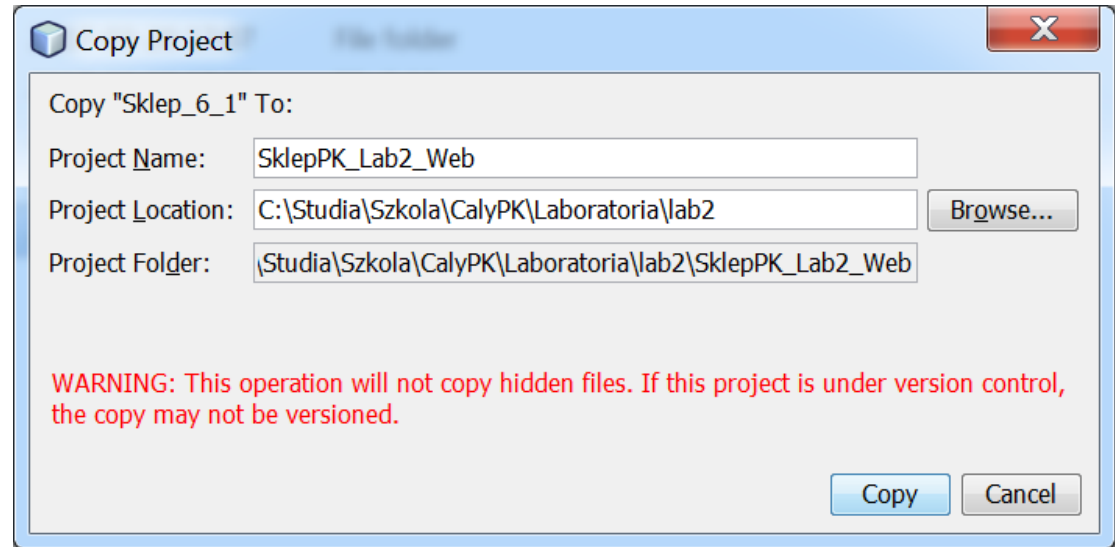
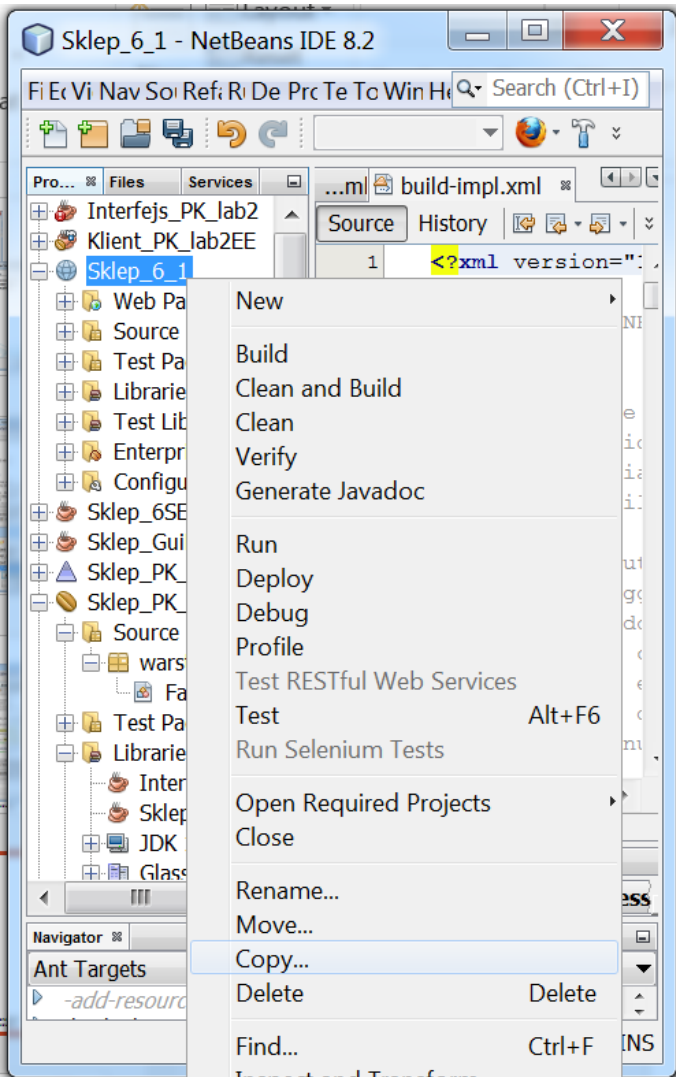
1.8. cd Kod interfejsu komponentu typu Session: **Fasada_warstwy_biznesowej_ejbRemote**

```
package warstwa_biznesowa_ejb;

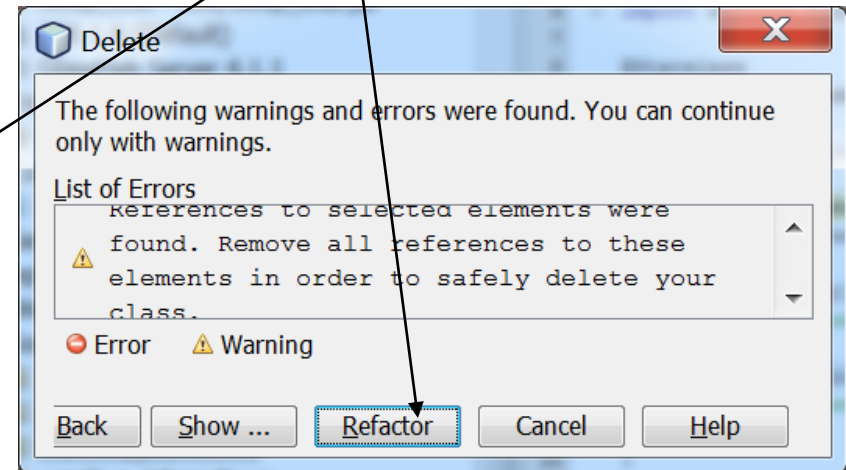
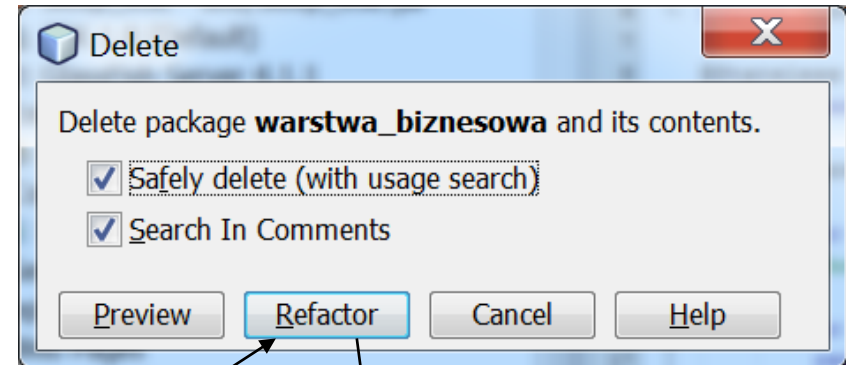
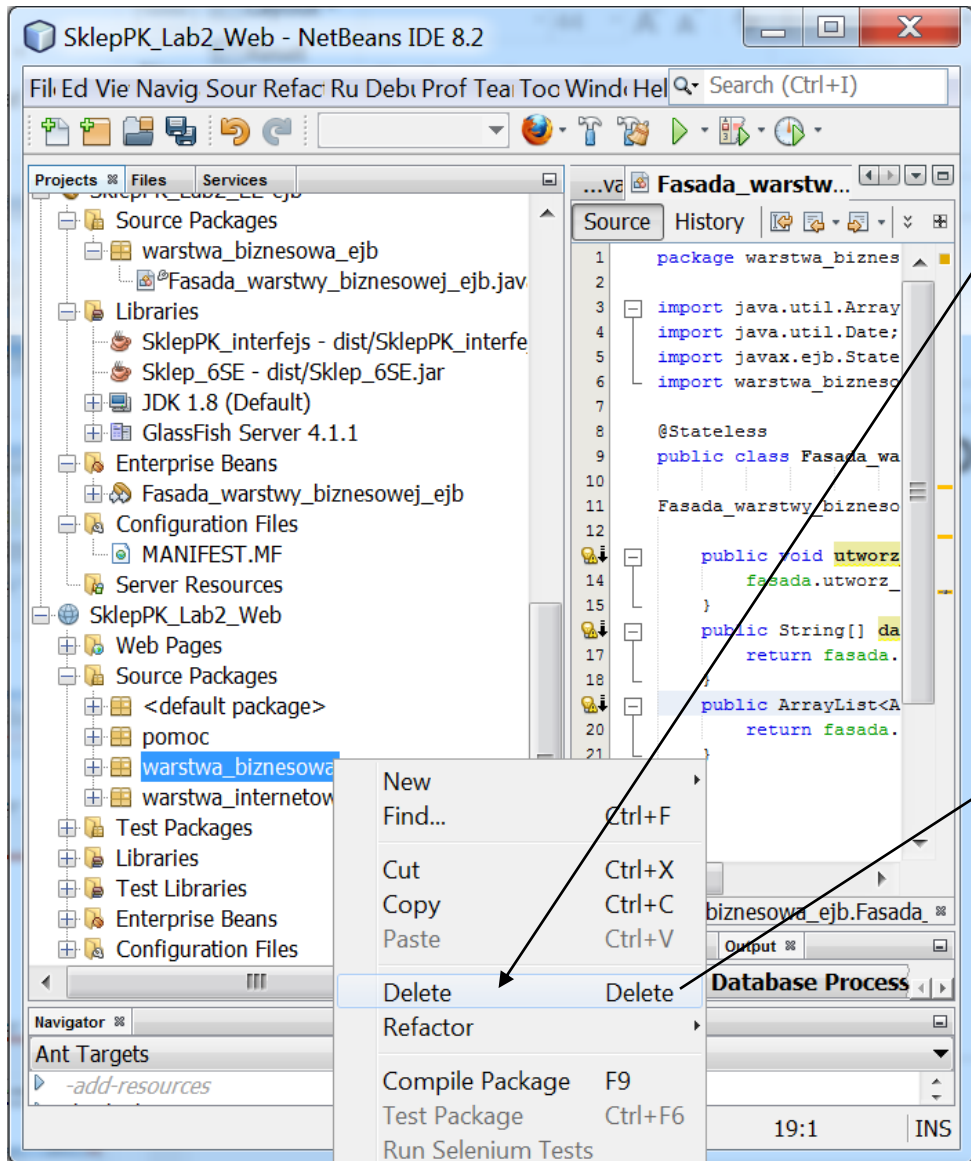
import java.util.ArrayList;
import java.util.Date;
import javax.ejb.Remote;

@Remote
public interface Fasada_warstwy_biznesowej_ejbRemote {
    public void utworz_produkt(String dane[], Date data);
    public String[] dane_produktu();
    public ArrayList<ArrayList<String>> items();
}
```

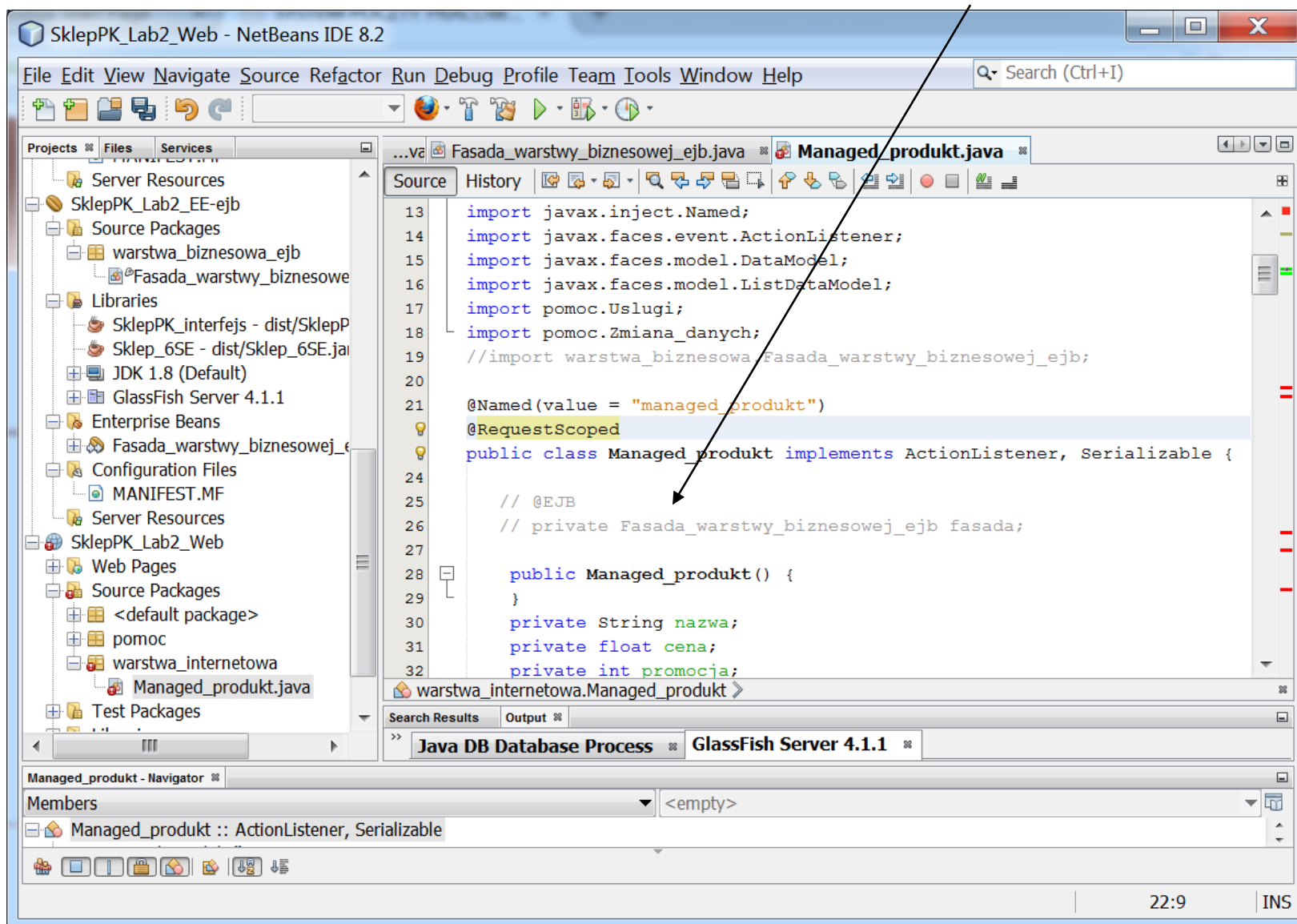
1.9. Dodanie do projektu EE warstwy klienta internetowego. Należy skopiować programu Sklep_6_1 z lab1 i nadać nową nazwę w polu **Project Name: SklepPK_Lab2_Web.**



1.10. W projekcie typu Java Web typu **Web Application** z p. 1.9 należy usunąć bezpiecznie pakiet **warstwa_biznesowa**.

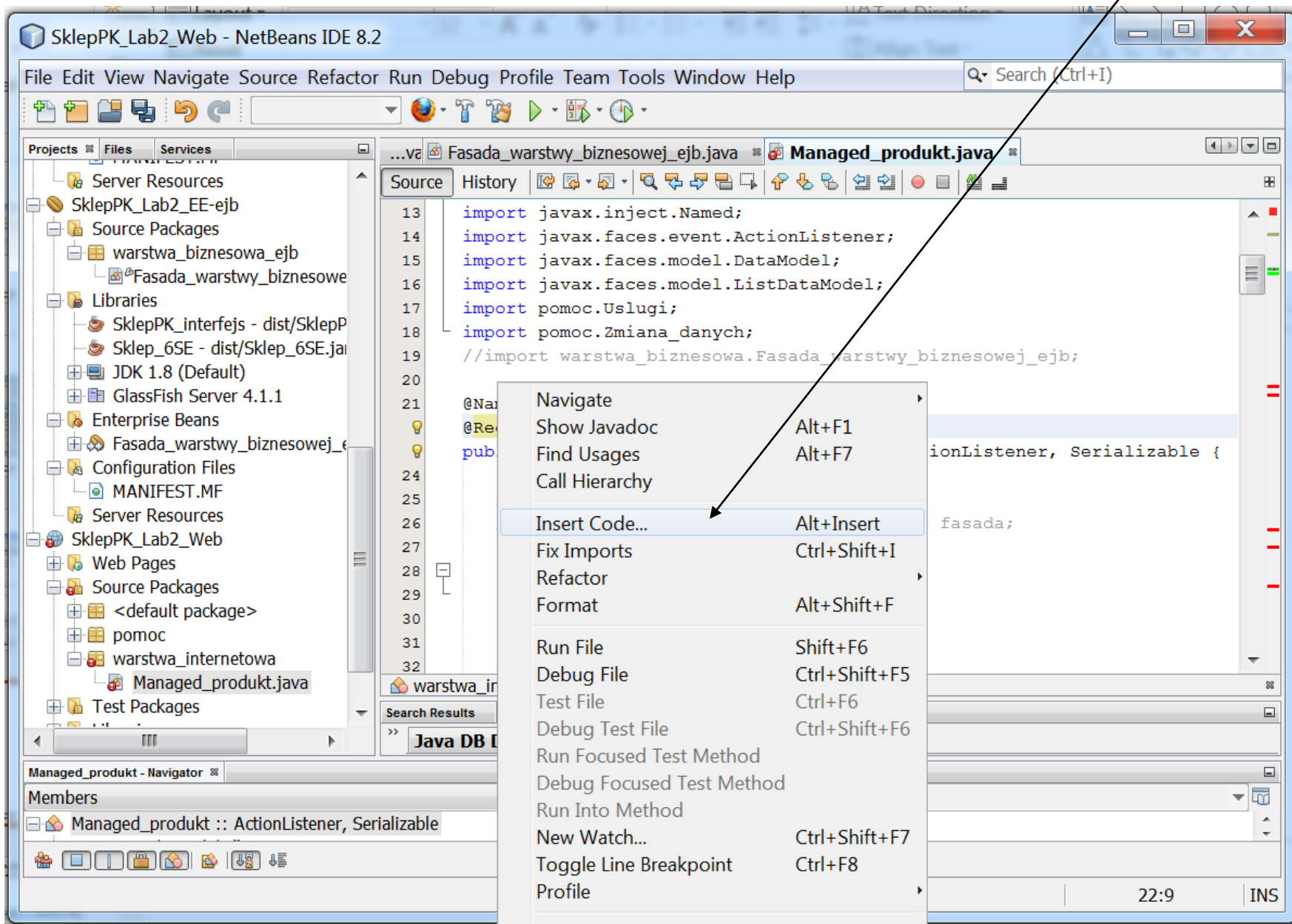


1.11. W komponencie typu Managed Bean o nazwie **Managed_produk**t należy usunąć z kodu dotychczasowe powiązanie z komponentem EJB.

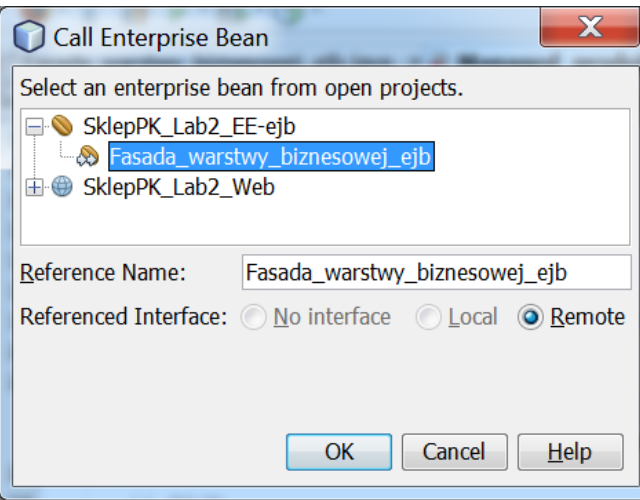


```
13 import javax.inject.Named;
14 import javax.faces.event.ActionListener;
15 import javax.faces.model.DataModel;
16 import javax.faces.model.ListDataModel;
17 import pomoc.Uslugi;
18 import pomoc.Zmiana_danych;
19 //import warstwa_biznesowa.Fasada_warstwy_biznesowej_ejb;
20
21 @Named(value = "managed_produk")
22 @RequestScoped
23 public class Managed_produk implements ActionListener, Serializable {
24
25     // @EJB
26     // private Fasada_warstwy_biznesowej_ejb fasada;
27
28     public Managed_produk() {
29     }
30     private String nazwa;
31     private float cena;
32     private int promocja;
```

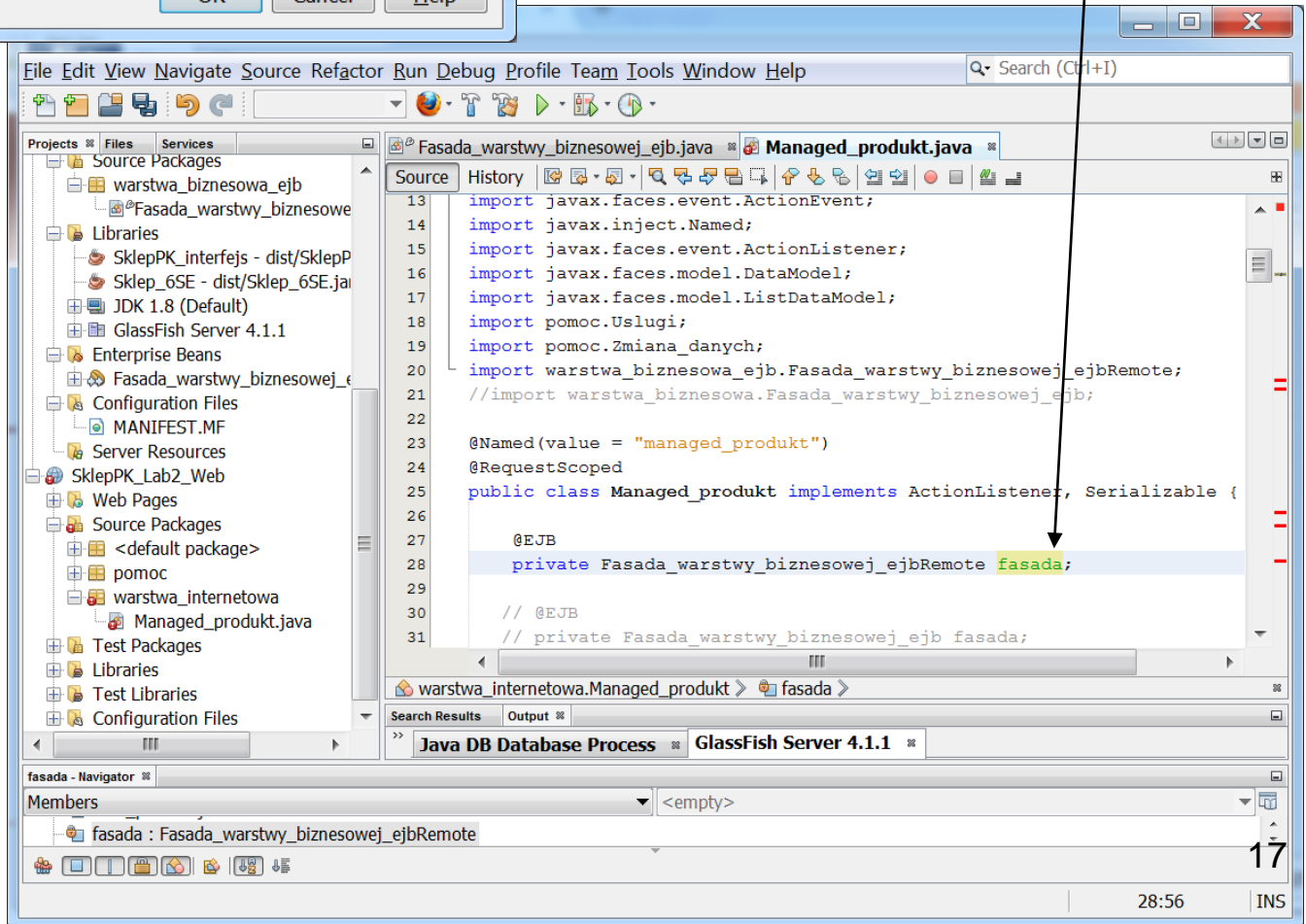
1.12. Należy teraz wstawić powiązanie z nowym komponentem typu EJB. W tym celu należy kliknąć prawym klawiszem myszy na kod klasy i wybrać pozycję **Insert Code...**



- Generate
- Constructor...
- Logger...
- Getter...
- equals() and hashCode()...
- toString()...
- Delegate Method...
- Override Method...
- Add Property...
- Call Enterprise Bean...
- Use Database...
- Send JMS Message...
- Send E-mail...
- Call Web Service Operation...
- Generate REST Client...



1.12.cd. Należy wybrać pozycję **Call Enterprise Bean..** i w kolejnym formularzu należy wybrać w projekcie EJB **SklepPK_Lab2_EE-ejb** komponent EJB typu **Session:**
Fasada_warstwy_biznesowej_ejbRemote i nadać mu nazwę **fasada**.



1.13. Wynik zmian – w celu usunięcia błędów należy zmodyfikować metody dostępu do dodanego komponentu: **Fasada_warstwy_biznesowej_ejbRemote**:

```
public Fasada_warstwy_biznesowej_ejbRemote getFasada()           { return fasada; }  
public void setFasada(Fasada_warstwy_biznesowej_ejbRemote fasada) { this.fasada = fasada; }
```

SklepPK_Lab2_Web - NetBeans IDE 8.2

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

Search (Ctrl+I)

Projects Files Services

Managed_produkkt.java Fasada_warstwy_biznesowej_ejb.java Fasada_warstw...

```
Source History  
18 import pomoc.Uslugi;  
19 import pomoc.Zmiana_danych;  
20 import warstwa_biznesowa_ejb.Fasada_warstwy_biznesowej_ejbRemote;  
21 //import warstwa_biznesowa.Fasada_warstwy_biznesowej_ejb;  
22  
23 @Named(value = "managed_produkkt")  
24 @RequestScoped  
25 public class Managed_produkkt implements ActionListener, Serializable  
26  
27     @EJB  
28     private Fasada_warstwy_biznesowej_ejbRemote fasada;  
29     // @EJB  
30     // private Fasada_warstwy_biznesowej_ejb fasada;  
31     public Fasada_warstwy_biznesowej_ejbRemote getFasada() {  
32         return fasada;  
33     }  
34     public void setFasada(Fasada_warstwy_biznesowej_ejbRemote fasada) {  
35         this.fasada = fasada;  
36     }
```

warstwa_internetowa.Managed_produkkt > setFasada >

Search Results Output

Java DB Database Process GlassFish Server 4.1.1

setFasada - Navigator

Members <empty>

setFasada(Fasada_warstwy_biznesowej_ejbRemote fasada)

34:2 | INS

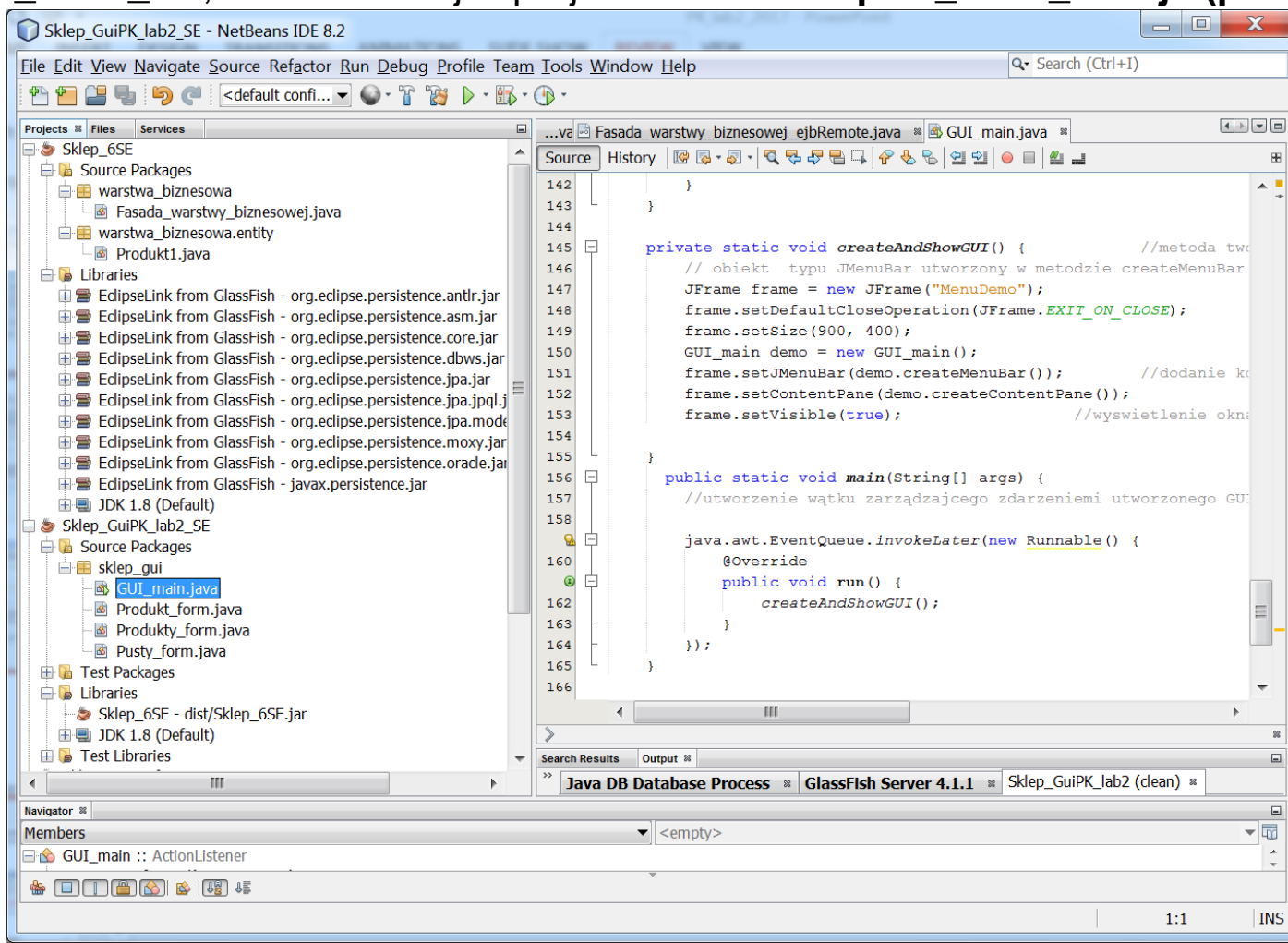
1.14. Dodanie warstwy klienta desktopowego Java EE typu Enterprise Application Client.

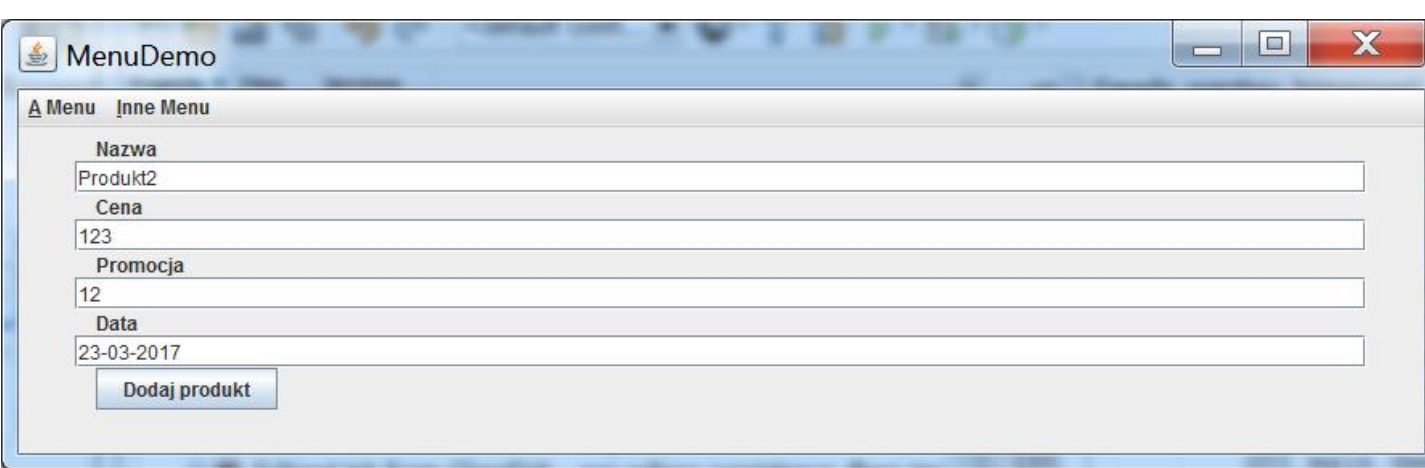
Pierwszym etapem jest utworzenie projektu na platformie Java SE

(http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/pojava/Lab6_2016.pdf).

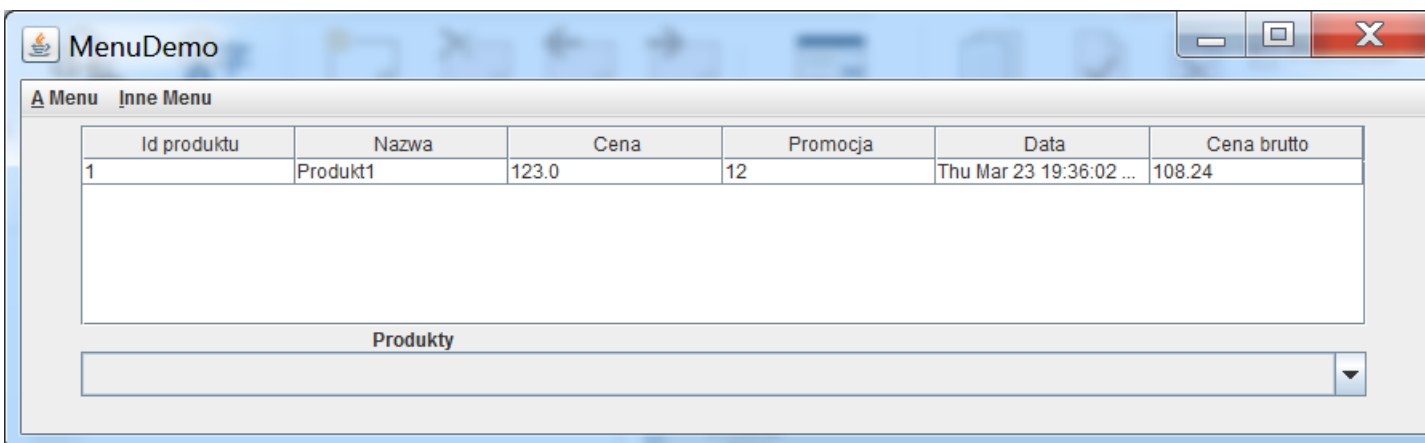
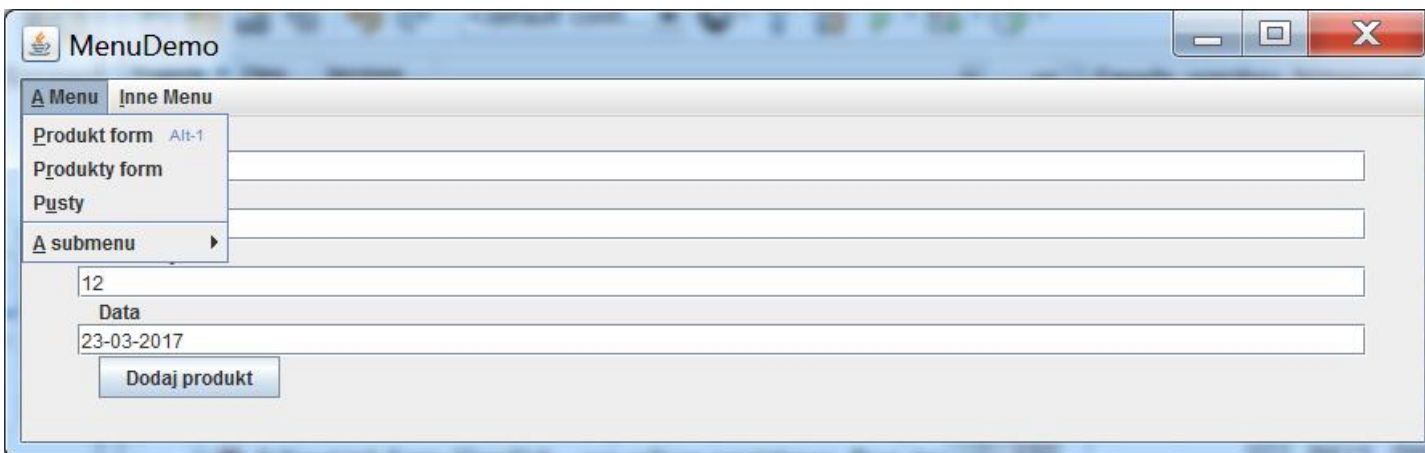
Logika biznesowa dla projektu **Sklep_GuiPK_lab2_SE** stanowiącego warstwę klienta w dwuwarstwowej aplikacji Java SE jest projekt **Sklep_6SE** – do pobrania:

(http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/javapk/Sklep_SE.rar). Jest to ten sam projekt, który dostarcza kod logiki biznesowej do projektu na platformie Java EE: **SklepPK_Lab2_EE**, umieszczonej w projekcie EJB: **SklepPK_Lab2_EE-ejb** (p.1.6).

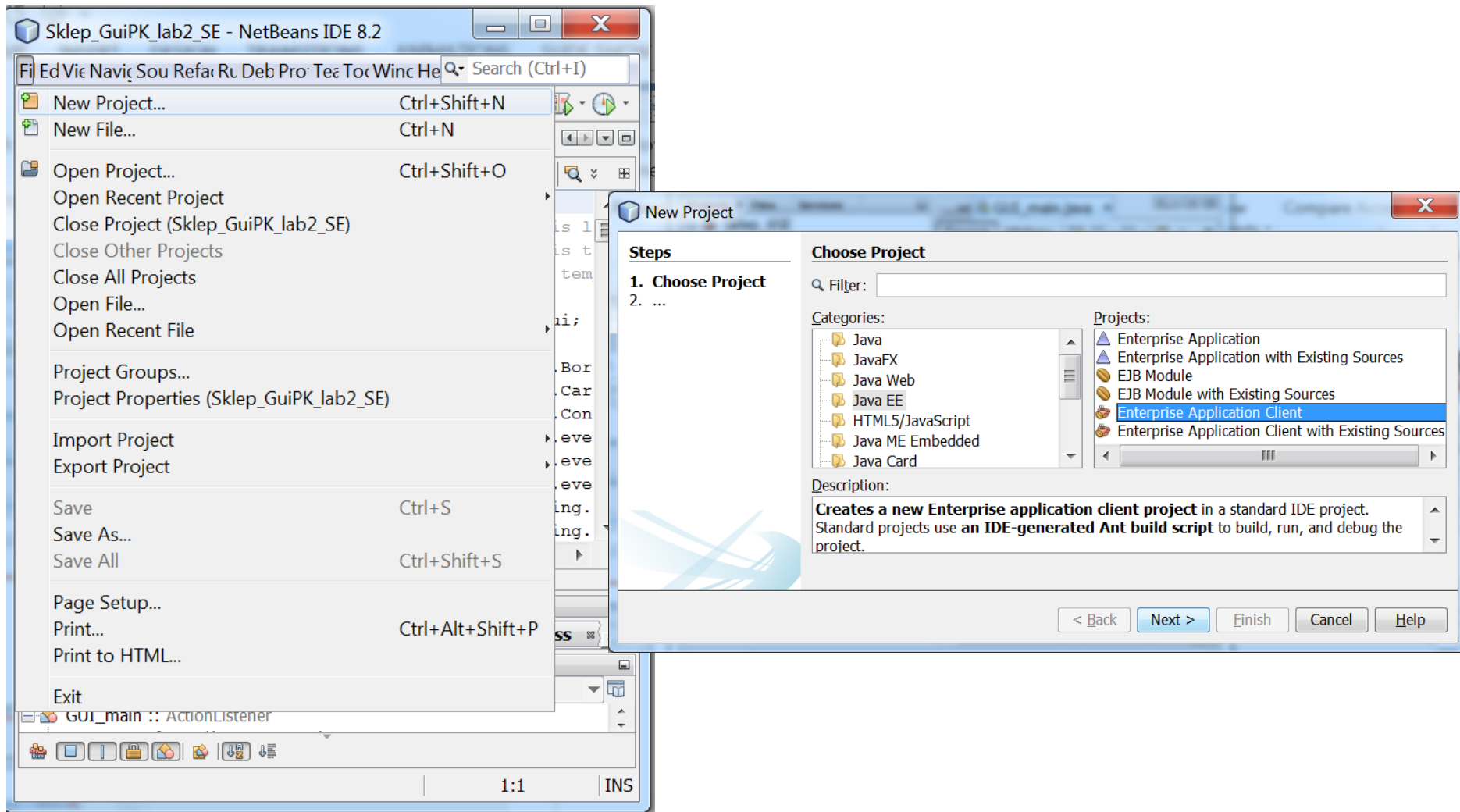




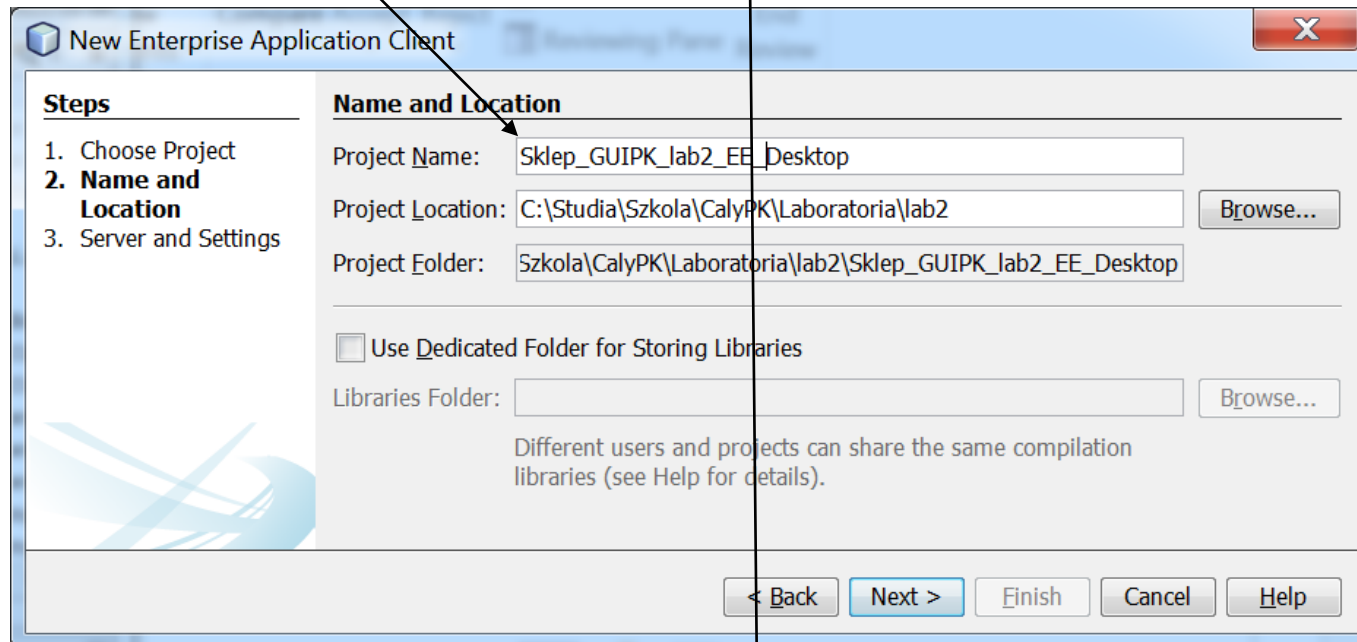
1.15. Wykonanie i uruchomienie projektu Java SE o architekturze dwuwarstwowej.



1.16. **Drugi etap dodania warstwy klienta desktopowego Java EE typu Enterprise Application Client.** Należy wybrać **File/New Project....** W kolejnym formularzu należy wybrać w Categories: **Java EE**, i w Projects: typu **Enterprise Application Client**. Należy kliknąć na klawisz **Next**.



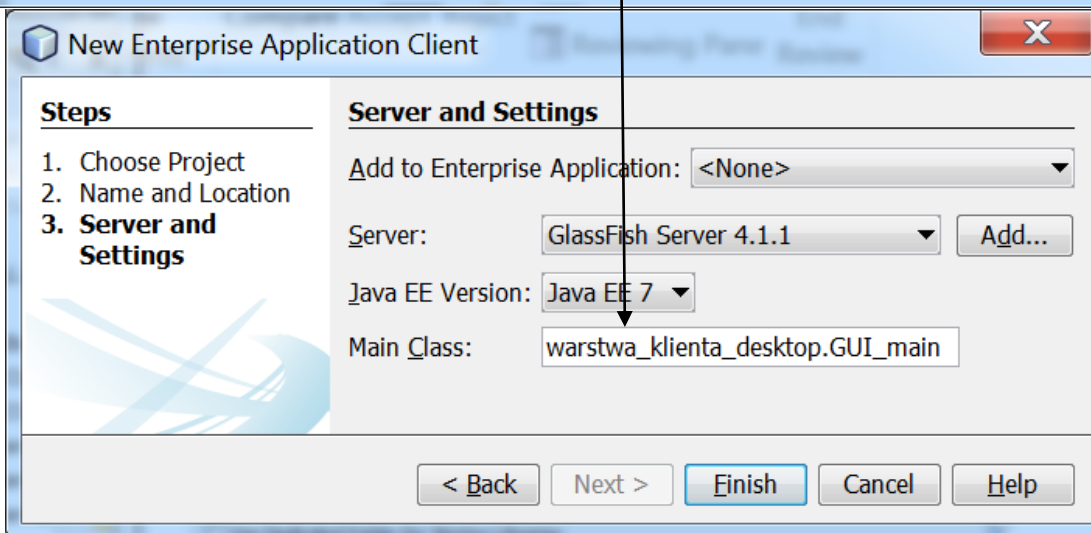
1.17. W kolejnym formularzu należy w polu **Project Name** wpisać nazwę projektu: **Sklep_GUIPK_lab2_EE_Desktop** i położenie w polu **Project Location**. W kolejnym formularzu w polu **Main Class** warto podać nazwę klasy z metodą main z projektu warstwy klienta SE: **Sklep_GuiPK_lab2_SE**, czyli **GUI_main**.



The screenshot shows the 'New Enterprise Application Client' wizard at the 'Name and Location' step. The 'Steps' list on the left indicates the current step is '2. Name and Location'. The main area contains the following fields:

- Project Name:** Sklep_GUIPK_lab2_EE_Desktop
- Project Location:** C:\Studia\Szkola\CalyPK\Laboratoria\lab2 (with a 'Browse...' button)
- Project Folder:** Szkola\CalyPK\Laboratoria\lab2\Sklep_GUIPK_lab2_EE_Desktop
- Use Dedicated Folder for Storing Libraries
- Libraries Folder:** (empty) (with a 'Browse...' button)

Below the fields is a note: 'Different users and projects can share the same compilation libraries (see Help for details).'. At the bottom are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

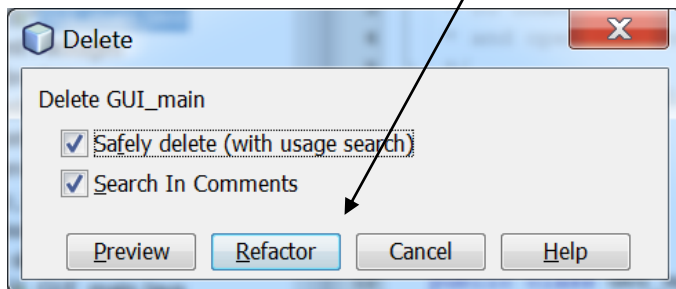
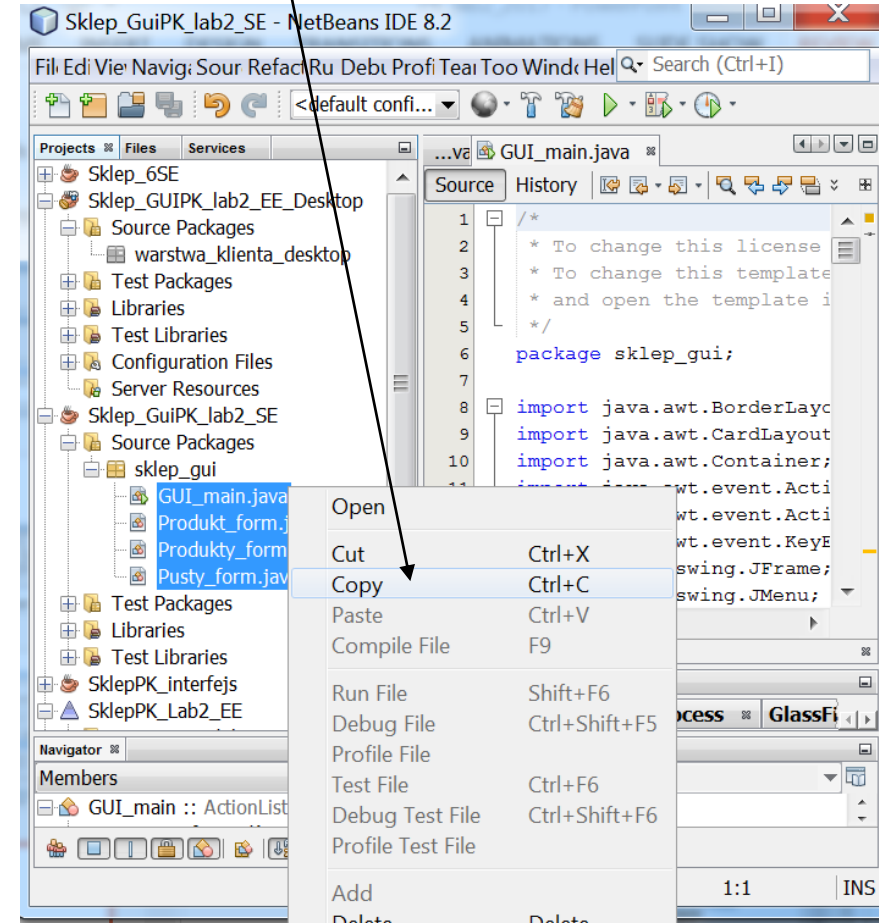
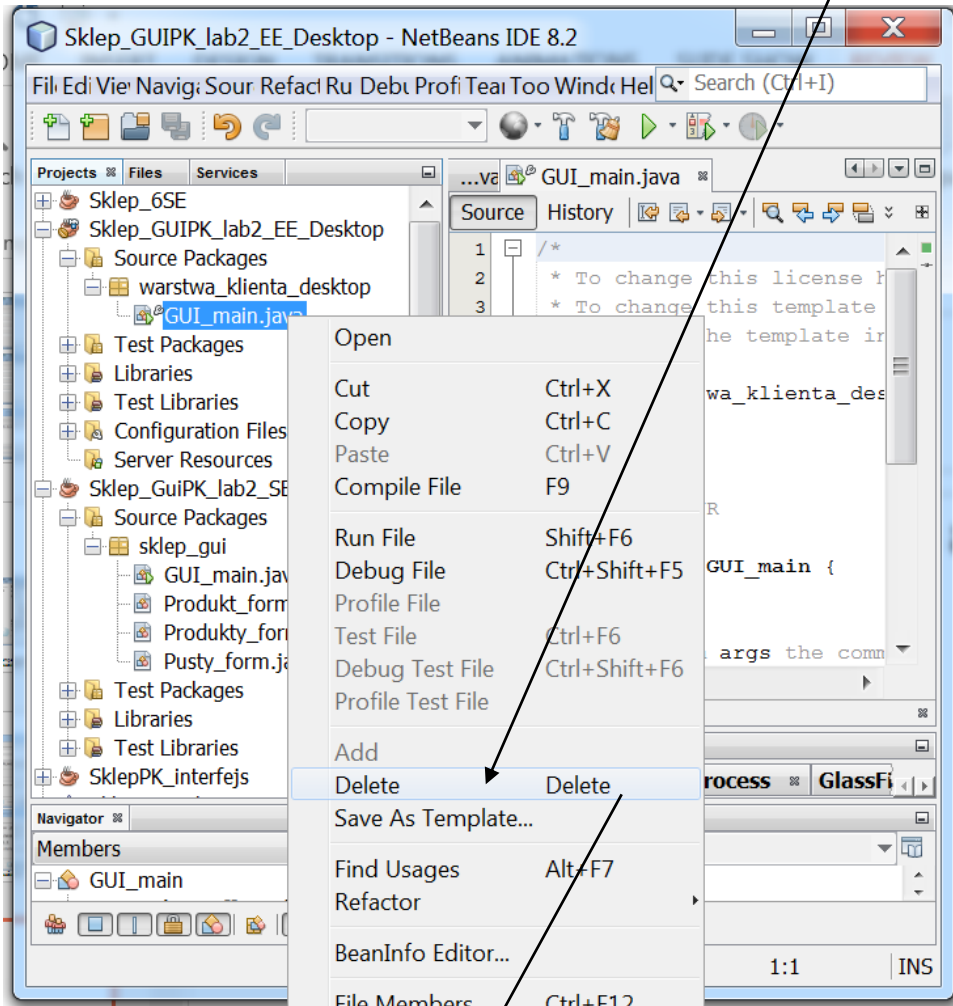


The screenshot shows the 'New Enterprise Application Client' wizard at the 'Server and Settings' step. The 'Steps' list on the left indicates the current step is '3. Server and Settings'. The main area contains the following fields:

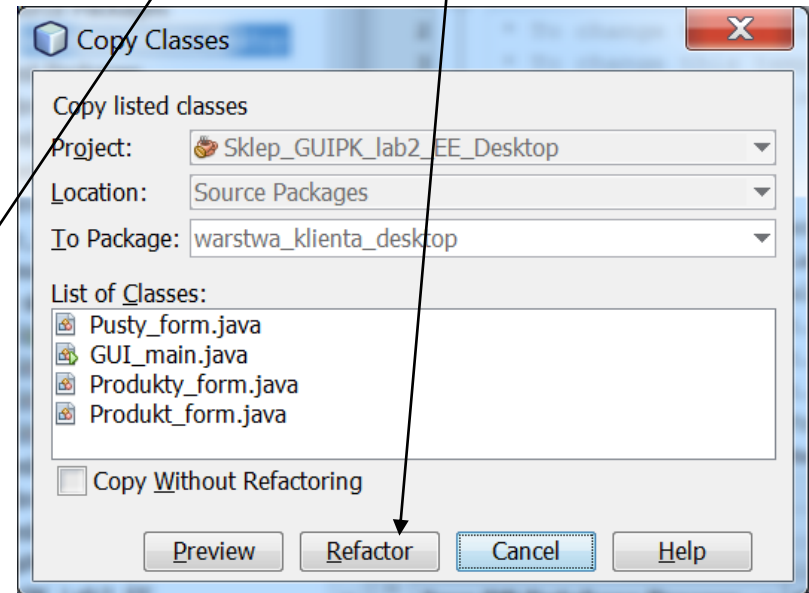
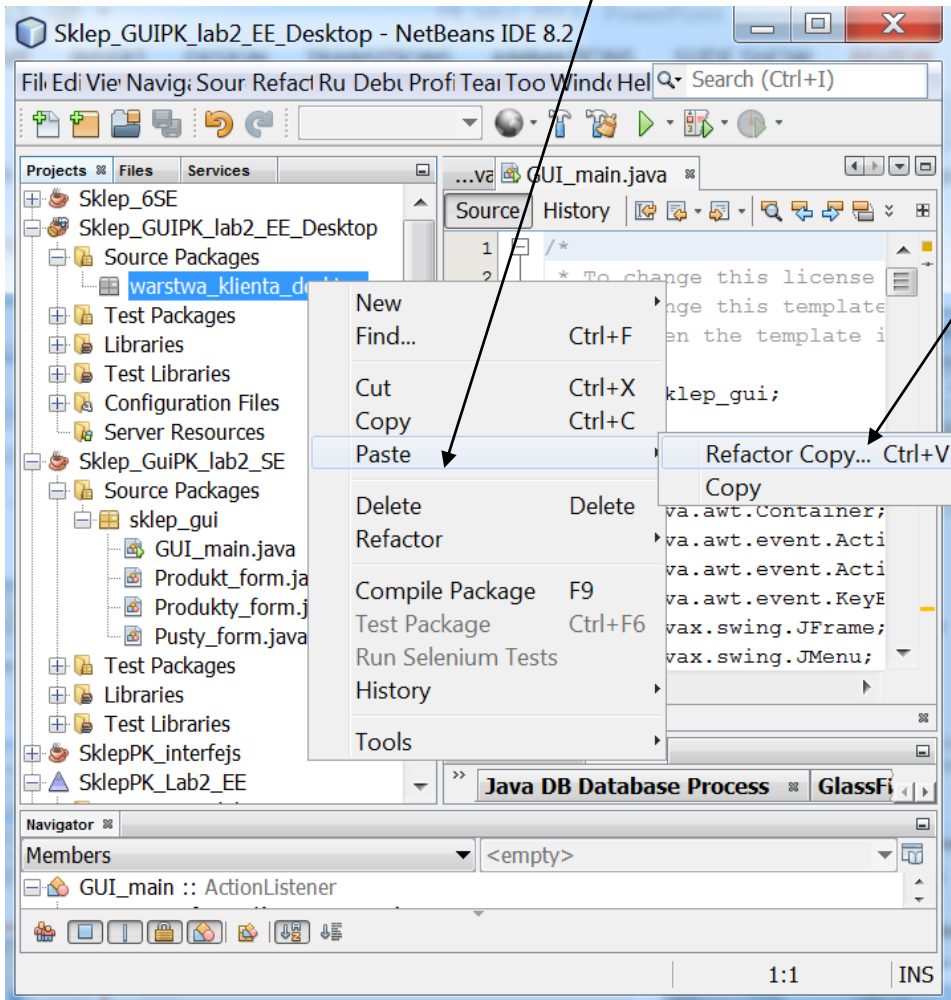
- Add to Enterprise Application:** <None> (dropdown)
- Server:** GlassFish Server 4.1.1 (dropdown) (with an 'Add...' button)
- Java EE Version:** Java EE 7 (dropdown)
- Main Class:** warstwa_klienta_desktop.GUI_main

At the bottom are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

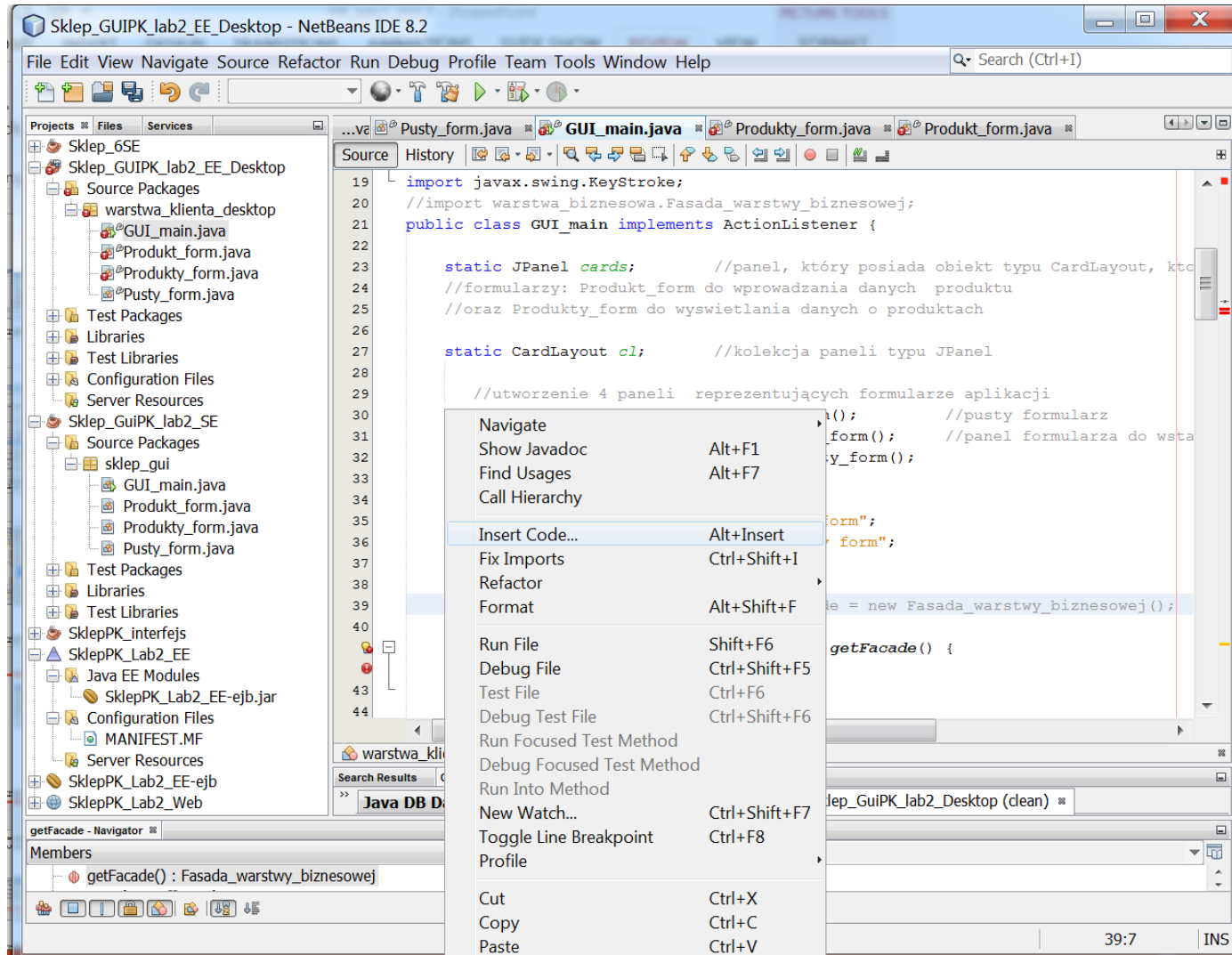
1.18. W nowym projekcie można teraz bezpiecznie usunąć klasę GUI_main. Następnie, należy skopiować pliki z formularzami z projektu klienta SE: **Sklep_GuiPK_lab2_SE**.



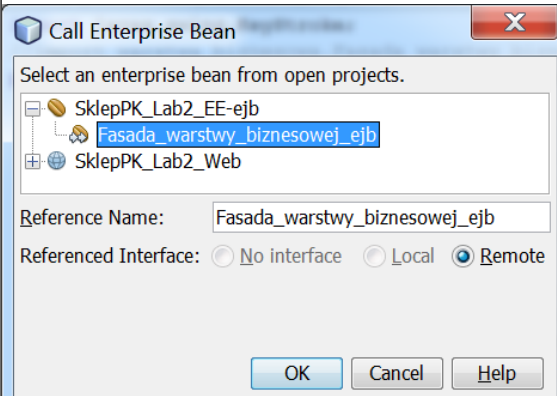
1.19. Następnie, należy je wkleić do nowego projektu Java EE typu **Enterprise Application Client: Sklep_GUIPK_lab2_EE_Desktop**, dokonując refaktoryzacji w zakresie przystosowania wklejanych plików do pakietu o nowej nazwie: **warstwa_klienta_desktop – Refactor Copy/Refactor**.



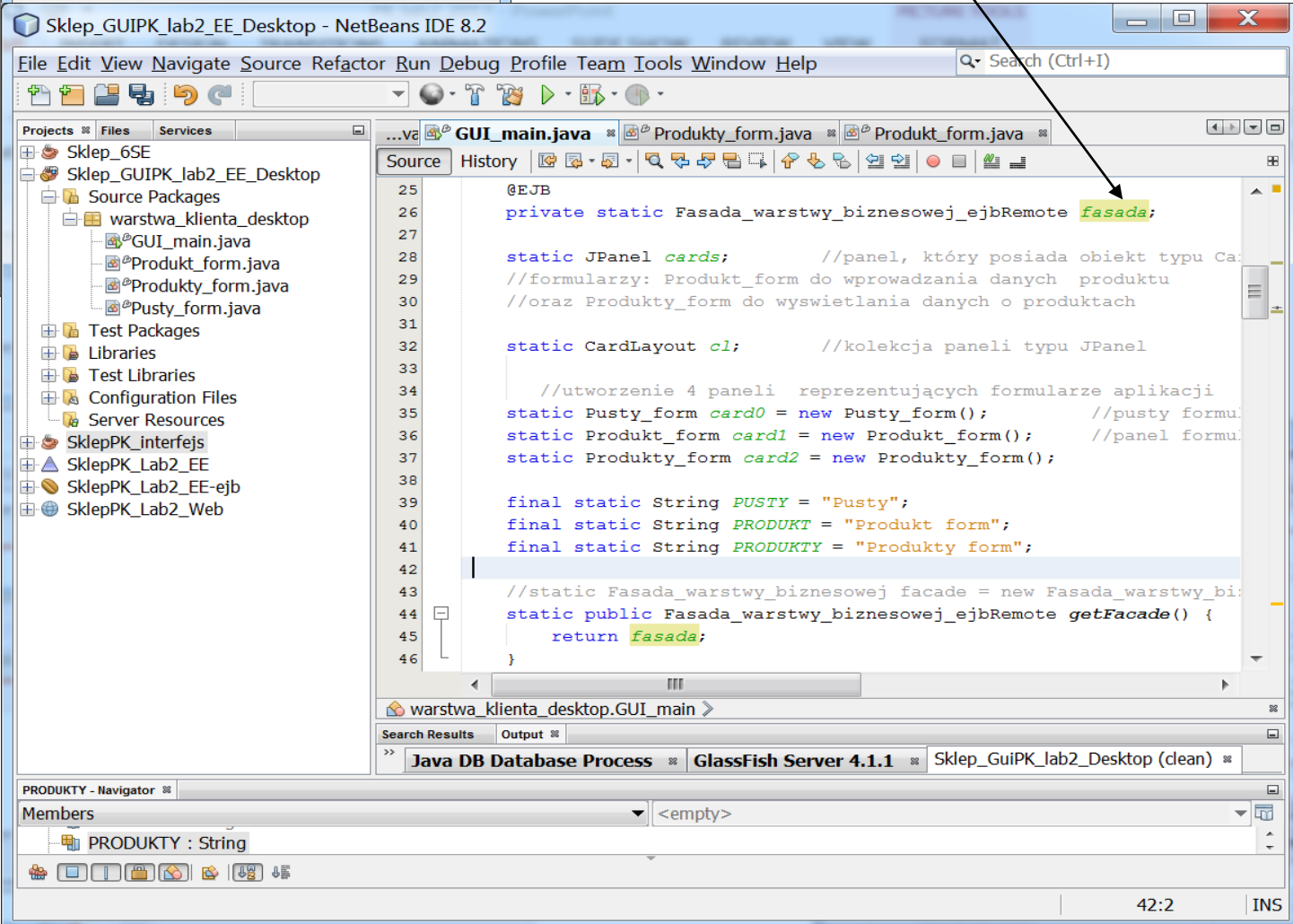
1.20. Należy w pliku GUI_main dodać połączenie z komponentem EJB typu Session: **Fasada_warstwy_biznesowej_ejb**, umieszczonym w projekcie EJB: **SklepPK_Lab2_EE-ejb**, w celu połączenia z logiką biznesową warstwy biznesowej aplikacji Java EE. W tym celu należy wybrać pozycję **Insert Code...**

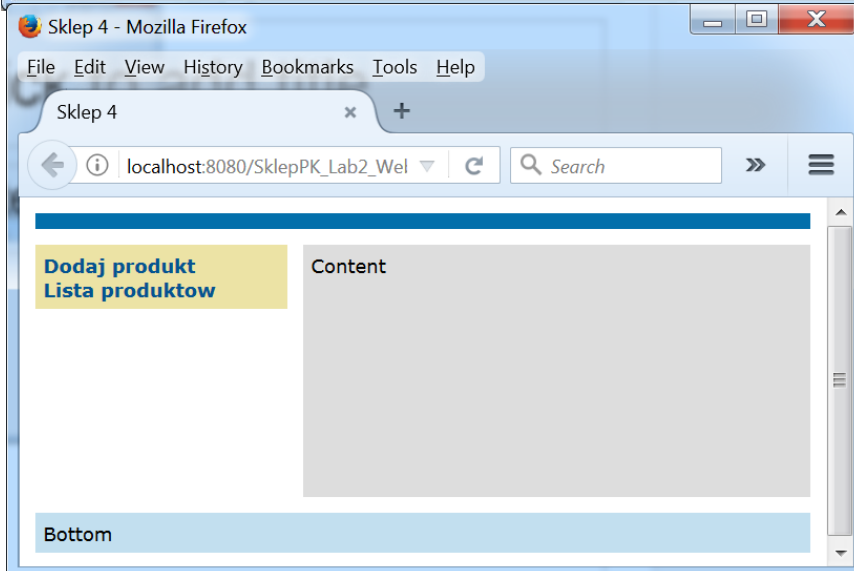
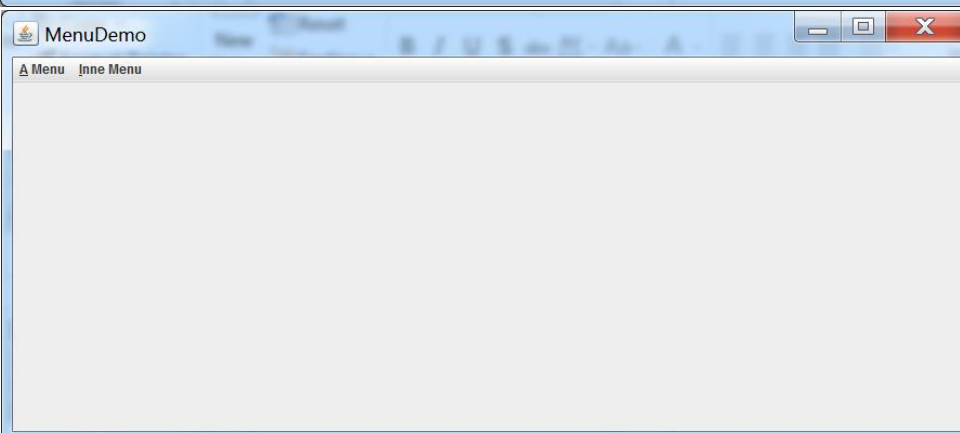
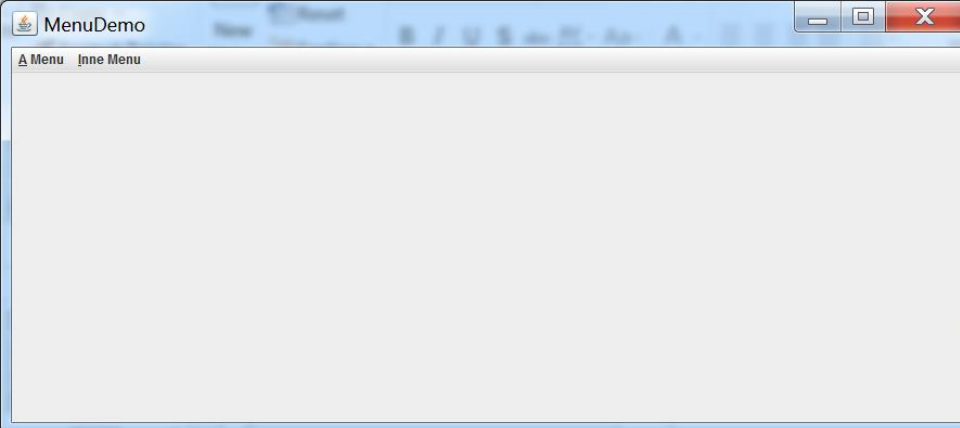


- Generate
- Constructor...
- Logger...
- Getter...
- Setter...
- Getter and Setter...
- toString()...
- Delegate Method...
- Override Method...
- Add Property...
- Call Enterprise Bean...**
- Use Database...
- Send JMS Message...
- Send E-mail...
- Call Web Service Operation...
- Generate REST Client...



1.21. Następnie, należy wybrać pozycję **Call Enterprise Bean...** dokonując wyboru komponentu EJB w projekcie EJB: **SklepPK_Lab2_EE-ejb**, czyli komponentu typu **Fasada_warstwy_biznesowej_ejb** o nazwie **fasada**.

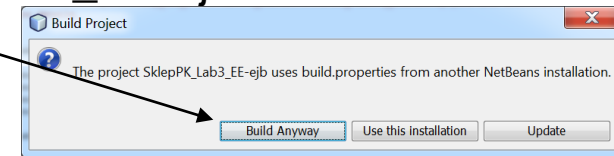




1.22. Uruchomienie projektu.

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

- 1) Sklep_6SE
- 2) SklepPK_interfejs
- 3) SklepPK_Lab2_EE-ebb



- 4) SklepPK_Lab2_Web
- 5) Sklep_GUIPK_lab2_EE_Desktop
- 6) SklepPK_Lab2_EE

Następnie, należy wykonać operację **Deploy** na projekcie **SklepPK_Lab2_EE**.

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

- 1) SklepPK_Lab2_Web
 - 2) Sklep_GUIPK_lab2_EE_Desktop
- W przykładzie uruchomiono dwie instancje aplikacji desktopowej i jedną internetową.

MenuDemo

A Menu Inne Menu

Nazwa
Produkt1

Cena
123

Promocja
12

Data
20-03-2017

Dodaj produkt

MenuDemo

A Menu Inne Menu

Nazwa
Produkt2

Cena
223

Promocja
22

Data
20-03-2017

Dodaj produkt

Wstawianie nowego produktu - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wstawianie nowego produktu x +

localhost:8080/SklepPK_Lab2

Dodaj produkt
Lista produktów

Podaj nazwe produktu Produkt3

Podaj cene netto produktu 330 zł

Podaj promocje produktu 33

Podaj date produkcji 18-03-2017

OK

Bottom

1.23. Proces wprowadzania danych w uruchomionych aplikacjach klienckich

MenuDemo

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt1	123.0	12	Mon Mar 20 20:25:10 CET 2017	108.24
2	Produkt2	223.0	22	Mon Mar 20 20:26:18 CET 2017	173.93999
3	Produkt3	330.0	33	Sat Mar 18 01:00:00 CET 2017	221.09999

Produkty

[2, Produkt2, 223.0, 22, Mon Mar 20 20:26:18 CET 2017, 173.93999]

A Menu Inne Menu

Id produktu	Nazwa	Cena	Promocja	Data	Cena brutto
1	Produkt1	123.0	12	Mon Mar 20 20:25:10 CET 2017	108.24
2	Produkt2	223.0	22	Mon Mar 20 20:26:18 CET 2017	173.93999
3	Produkt3	330.0	33	Sat Mar 18 01:00:00 CET 2017	221.09999

Produkty

1.24. Każda z aplikacji klienckich korzysta z danych umieszczonych w tym samym komponencie EJB typu Session – rodzaj Stateless.

Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Lista produktow

localhost:8080/SklepPK_Lab2_Web/faces/warstwa_interneto

Dodaj produkt
Lista produktow

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	123.0	12	Mon Mar 20 20:25:10 CET 2017	108.24
2	Produkt2	223.0	22	Mon Mar 20 20:26:18 CET 2017	173.93999
3	Produkt3	330.0	33	Sat Mar 18 01:00:00 CET 2017	221.09999

Powrot

Bottom

2. Należy zmodyfikować rodzaj komponentu EJB typu session w projekcie EJB: **SklepPK_Lab2_EE na Stateful. Należy opisać różnice w zachowaniu aplikacji klienckich: desktopowej i internetowej wg wykładów:**

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/javapk/PK_2.pdf

http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/javapk/PK_5_2017.pdf