

Internet Engineering

**Tomasz Babczyński, Zofia Kruczkiewicz
Tomasz Kubik**

**Information systems modelling – UML and
service description languages
Laboratory 3**

Choose yourself and new technologies



HUMAN CAPITAL
HUMAN – BEST INVESTMENT!



Wrocław University of Technology

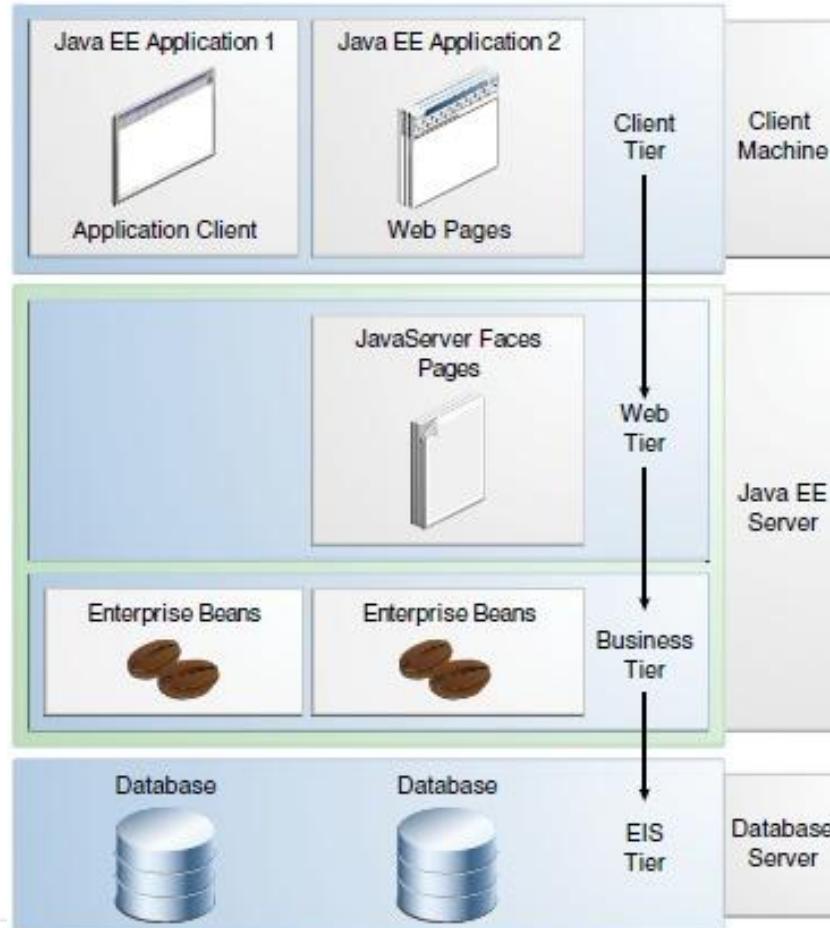
EUROPEAN
SOCIAL FUND





Exemplary multitiered Information System

(Java EE 7 – Tutorial Java EE 7)





Steps for building the client tier of a multitier application

1. **Create the Java EE project - right click File/New Project**
2. **Creation the library project for interface of EJB**
3. **Adding the EJB as the facade**
4. **Creation the library project for code from lab1-2**
5. **Adding the library project (p.4) to the EJB module of EE project and its interface**
6. **Creation of EE client tier**
7. **Build, deploy and run the created software**

Example (Java EE & Java Web Learning Trail)

<https://netbeans.org/kb/docs/javaee/entappclient.html>

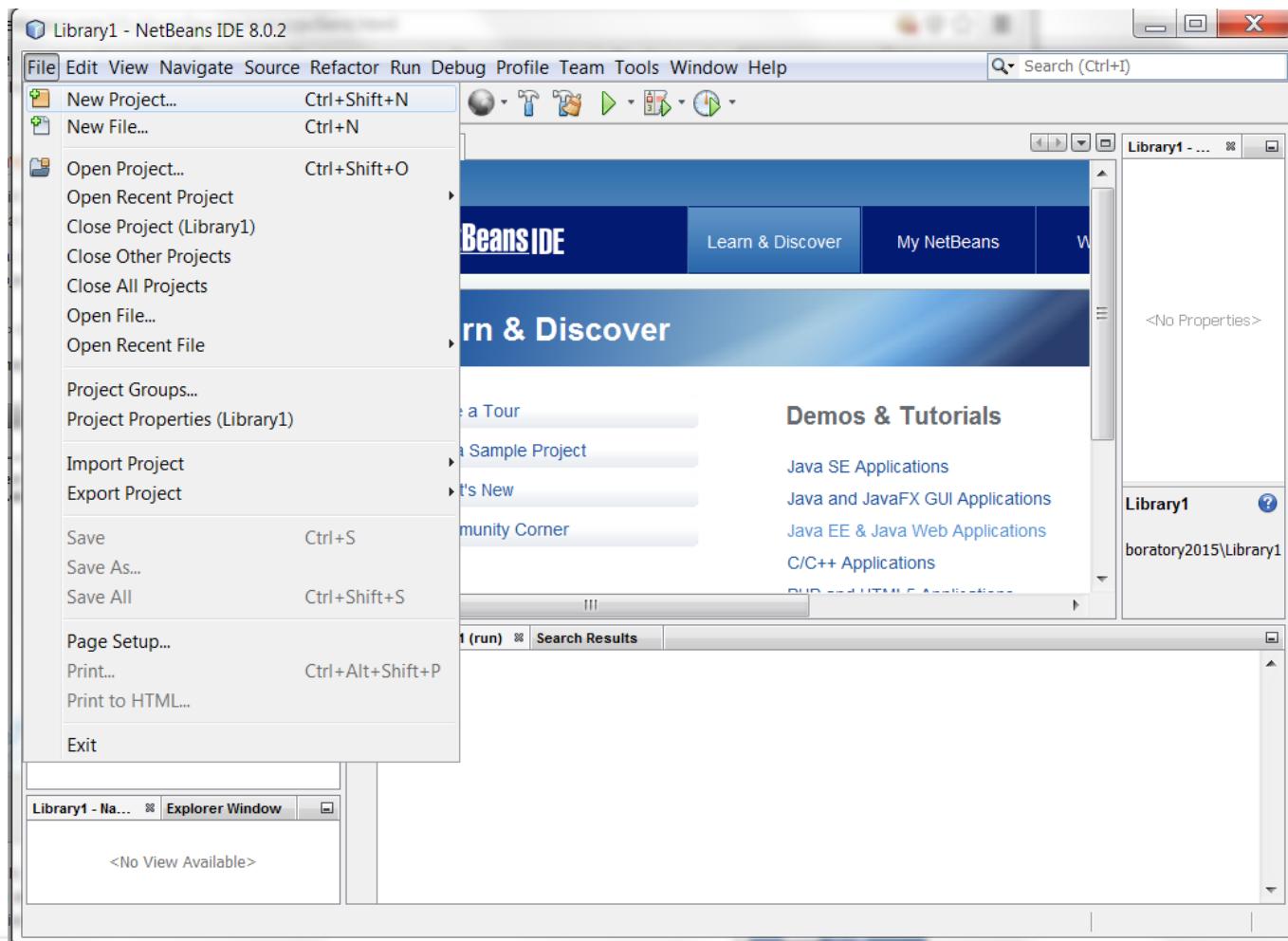
from

https://netbeans.org/kb/trails/java-ee.html?utm_source=netbeans&utm_campaign=welcomepage





1. / 1.1.Create the Java EE project - right click File/New Project





1.2. Select Java EE\ Enterprise Application and click Next

New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- UML
- Java Web
- Java EE**
- HTML5
- Java ME Embedded
- Java Card
- Maven
- PHP
- Groovy

Projects:

- Enterprise Application
- Enterprise Application with Existing Sources
- EJB Module
- EJB Module with Existing Sources
- Enterprise Application Client
- Enterprise Application Client with Existing S

Description:

Creates a new enterprise application in a standard project. You can also create an EJB module project and Web application project in the enterprise application. A standard project uses an **IDE-generated Ant build script** to build and run your projects.

< Back **Next >** Finish Cancel Help





1.3. Name and Location of EE project and next

New Enterprise Application

X

Steps	Name and Location
1. Choose Project	Project Name: <input type="text" value="Library1_EJB1"/>
2. Name and Location	Project Location: <input type="text" value="C:\EnglishLecture\Kruczkieicz\Laboratory2015"/> <input type="button" value="Browse..."/>
3. Server and Settings	Project Folder: <input type="text" value="nglishLecture\Kruczkieicz\Laboratory2015\Library1_EJB1"/>
	<input type="checkbox"/> Use Dedicated Folder for Storing Libraries Libraries Folder: <input type="text"/> <input type="button" value="Browse..."/> <small>Different users and projects can share the same compilation libraries (see Help for details).</small>

< Back Finish Cancel Help





1.4. Server and Settings of EE project and next

New Enterprise Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**

Server and Settings

Server: GlassFish Server 4.1

Java EE Version: Java EE 7

Create EJB Module: Library1_EJB1-ejb

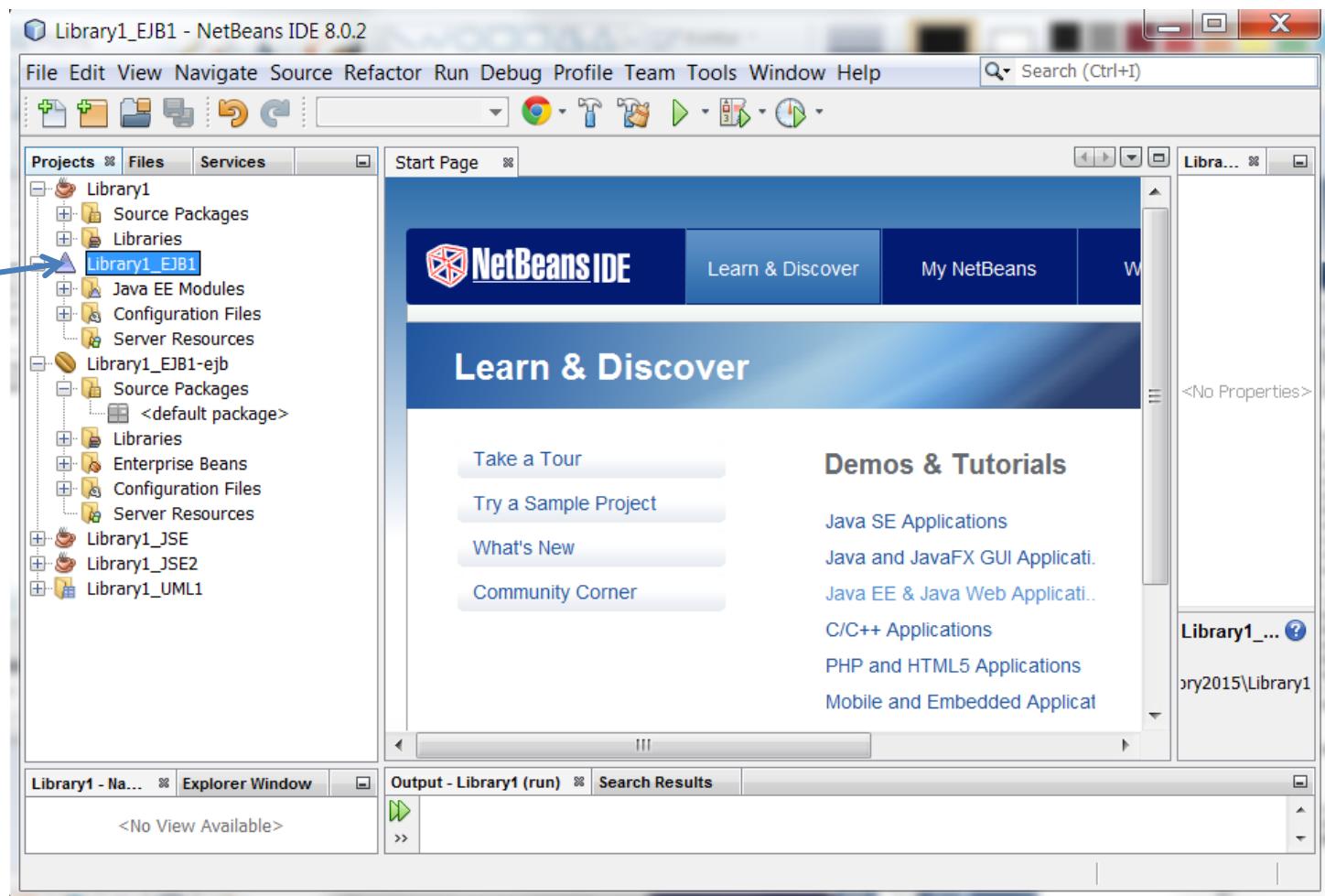
Create Web Application Module: Library1_EJB1-war

< Back



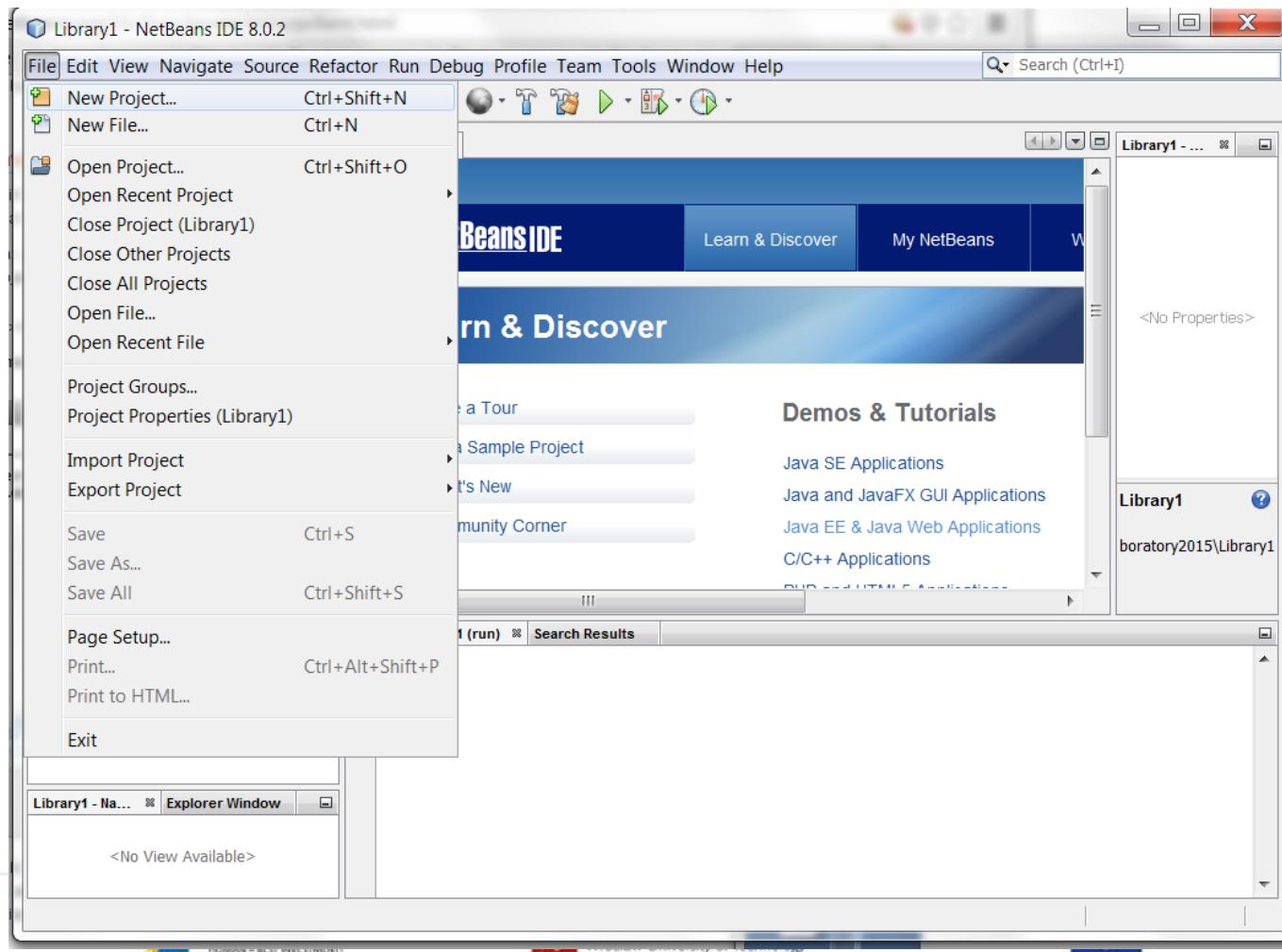


1.5. The new EE project



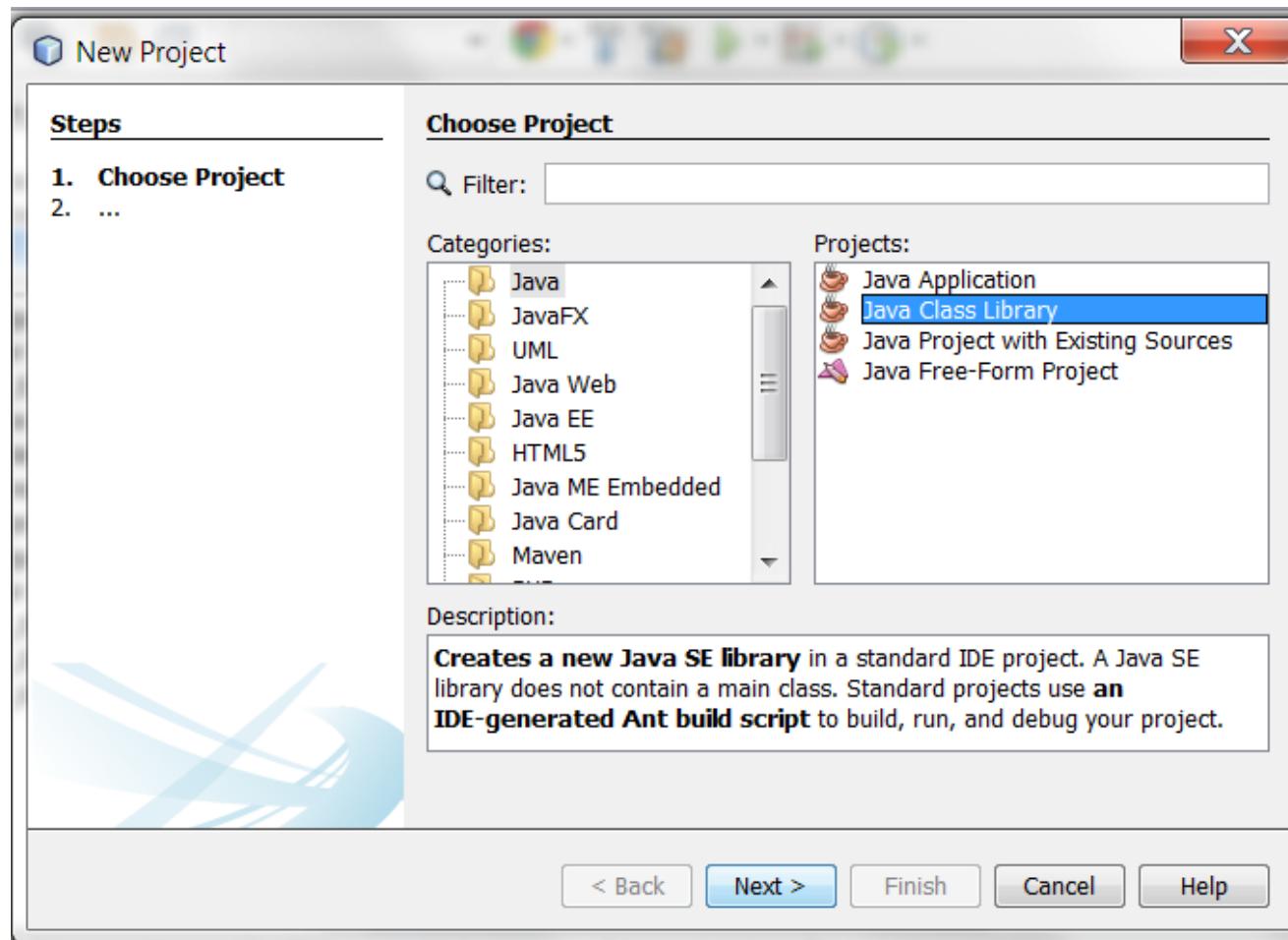


2. / 2.1. Creation the library project for interface of EJB





2.2. Creation the library project for interface of EJB – select Java/Java Class Library items





2.3. Creation the library project for interface of EJB – fill the Project Name and select the Project Location

New Java Class Library

X

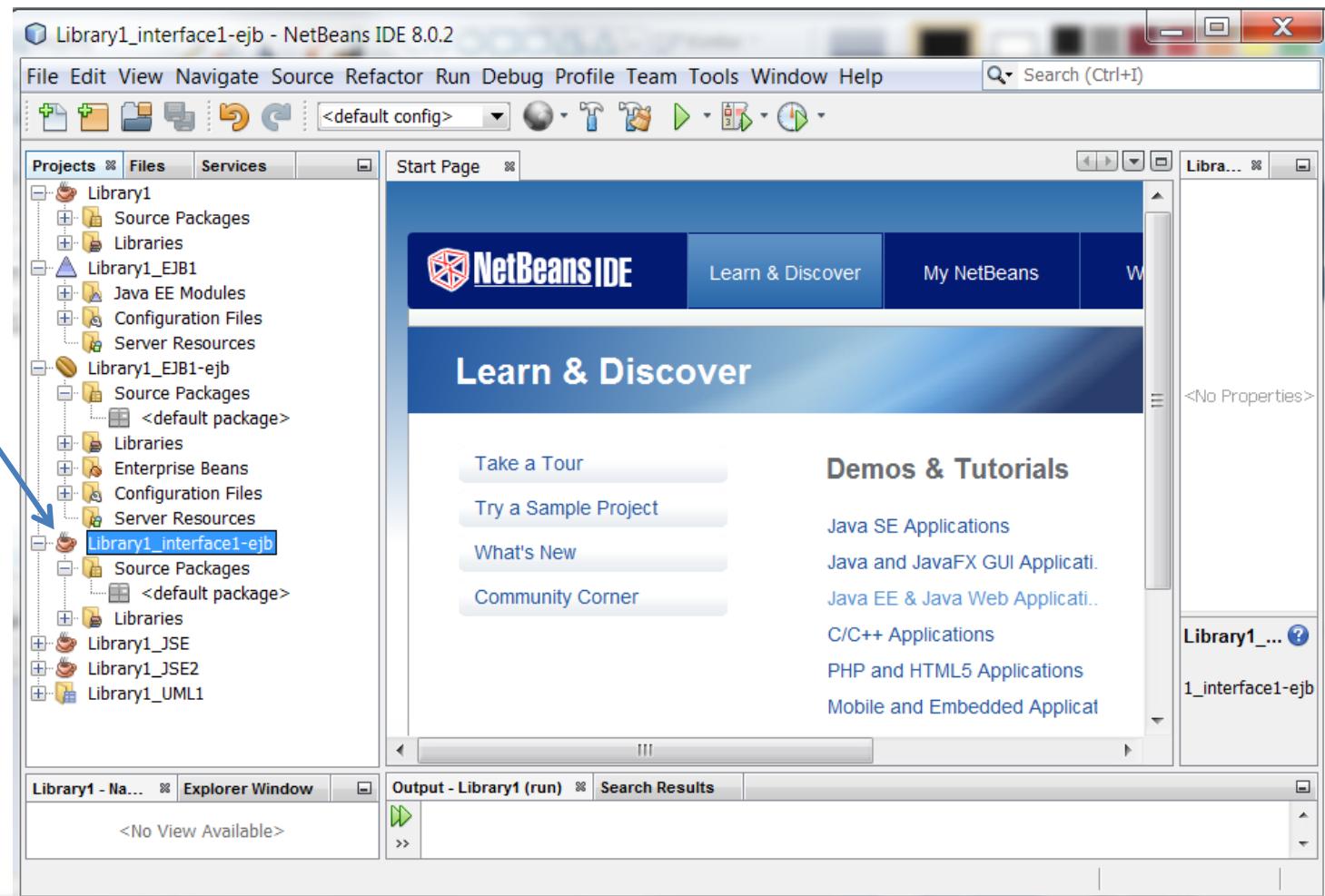
Steps	Name and Location
1. Choose Project 2. Name and Location	<p>Project Name: <input type="text" value="Library1_interface1-ejb"/></p> <p>Project Location: <input type="text" value="C:\EnglishLecture\Kruckiewicz\Laboratory2015"/> <input type="button" value="Browse..."/></p> <p>Project Folder: <input type="text" value="uczkiewicz\Laboratory2015\Library1_interface1-ejb"/></p> <p><input type="checkbox"/> Use Dedicated Folder for Storing Libraries Libraries Folder: <input type="text"/> <input type="button" value="Browse..."/></p> <p>Different users and projects can share the same compilation libraries (see Help for details).</p>

< Back



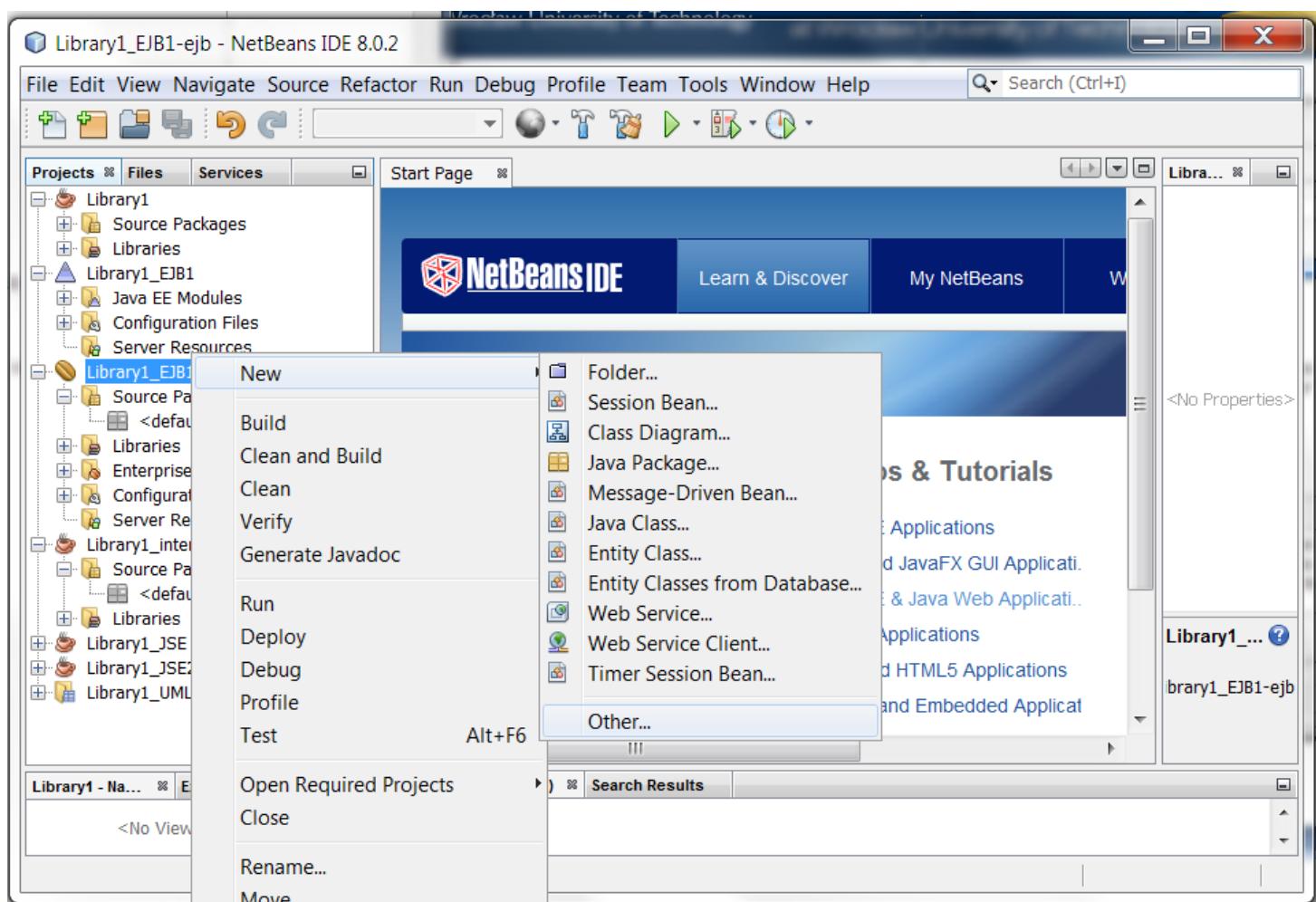


2.4. The new empty library project





3. / 3.1. Adding the EJB as the Session Facade – Right click/New/Other





3.2. Adding the EJB as the Session Facade – select Enterprise JavaBeans/ Session Bean

New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: Library1_EJB1-ejb

Filter:

Categories:

- UML
- Bean Validation
- Enterprise JavaBeans
- Contexts and Dependenc
- Java
- JavaBeans Objects

File Types:

- Session Bean
- Timer Session Bean
- Message-Driven Bean
- Service Locator
- Caching Service Locator
- Session Beans For Entity Classes
- Standard Deployment Descriptor

Description:

Creates an empty Session Enterprise JavaBean (EJB) component. A session bean is typically used to encapsulate business logic or enterprise resources. This template creates the Java classes for a single session bean and registers the

< Back Next > Finish Cancel Help





3.3. Adding the EJB as the Session Facade – fill the EJB Name, name of package, select the Session Type and create interface for Remote EJB

New Session Bean

Steps

1. Choose File Type
2. Name and Location

Name and Location

EJB Name:

Project:

Location:

Package:

Session Type:

Stateless

Stateful

Singleton

Create Interface:

Local

Remote in project:

< Back





3.4. The new empty EJB and its empty interface

```
/*
 * To change this license header, choose License Headers in
 * Project Properties. To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package business_tier;

import javax.ejb.Stateless;

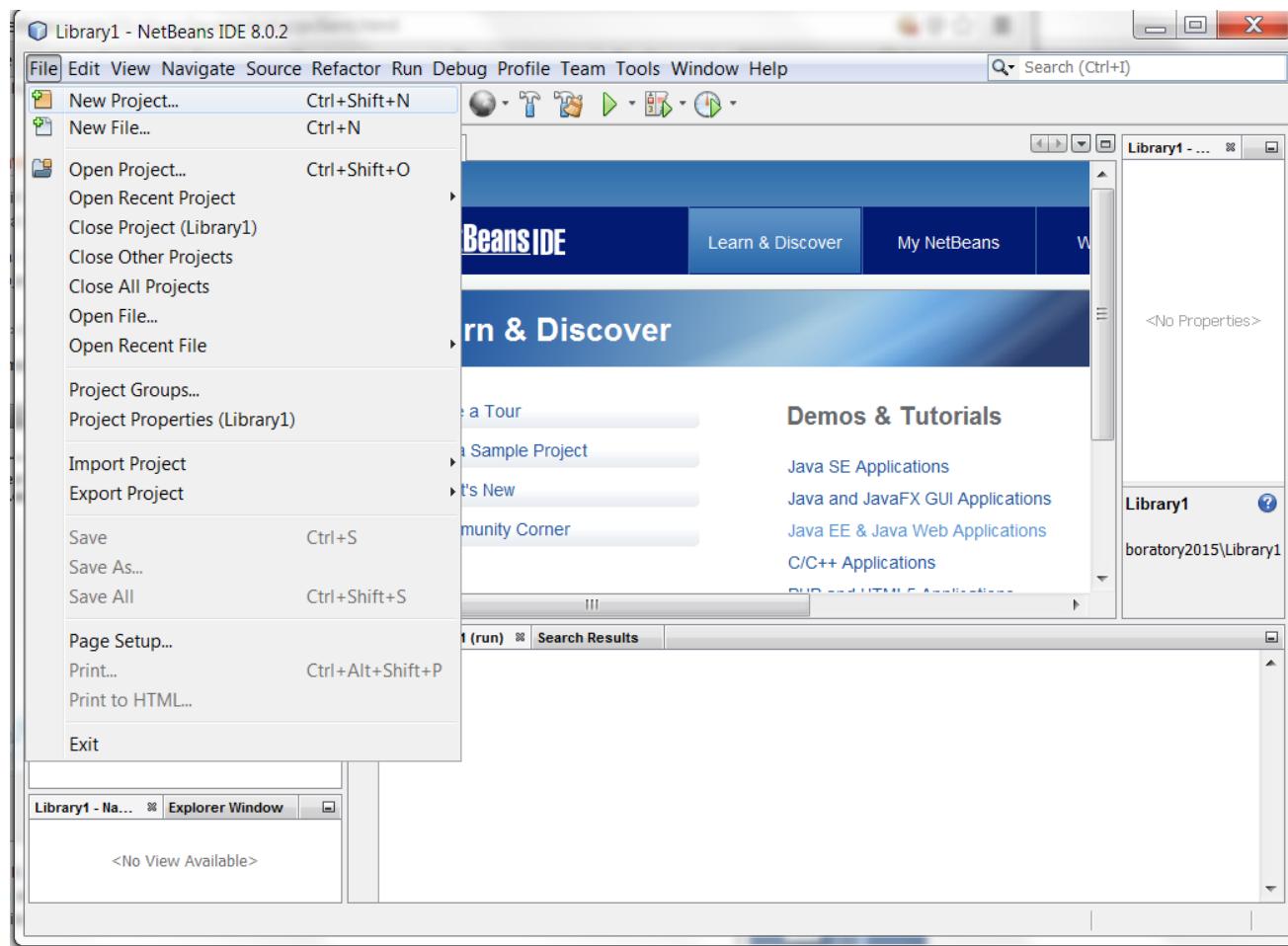
/**
 *
 * @author Zofia
 */
@Stateless
public class Facade implements FacadeRemote {

    // Add business logic below. (Right-click in editor and
    // "Insert Code > Add Business Method")
}
```



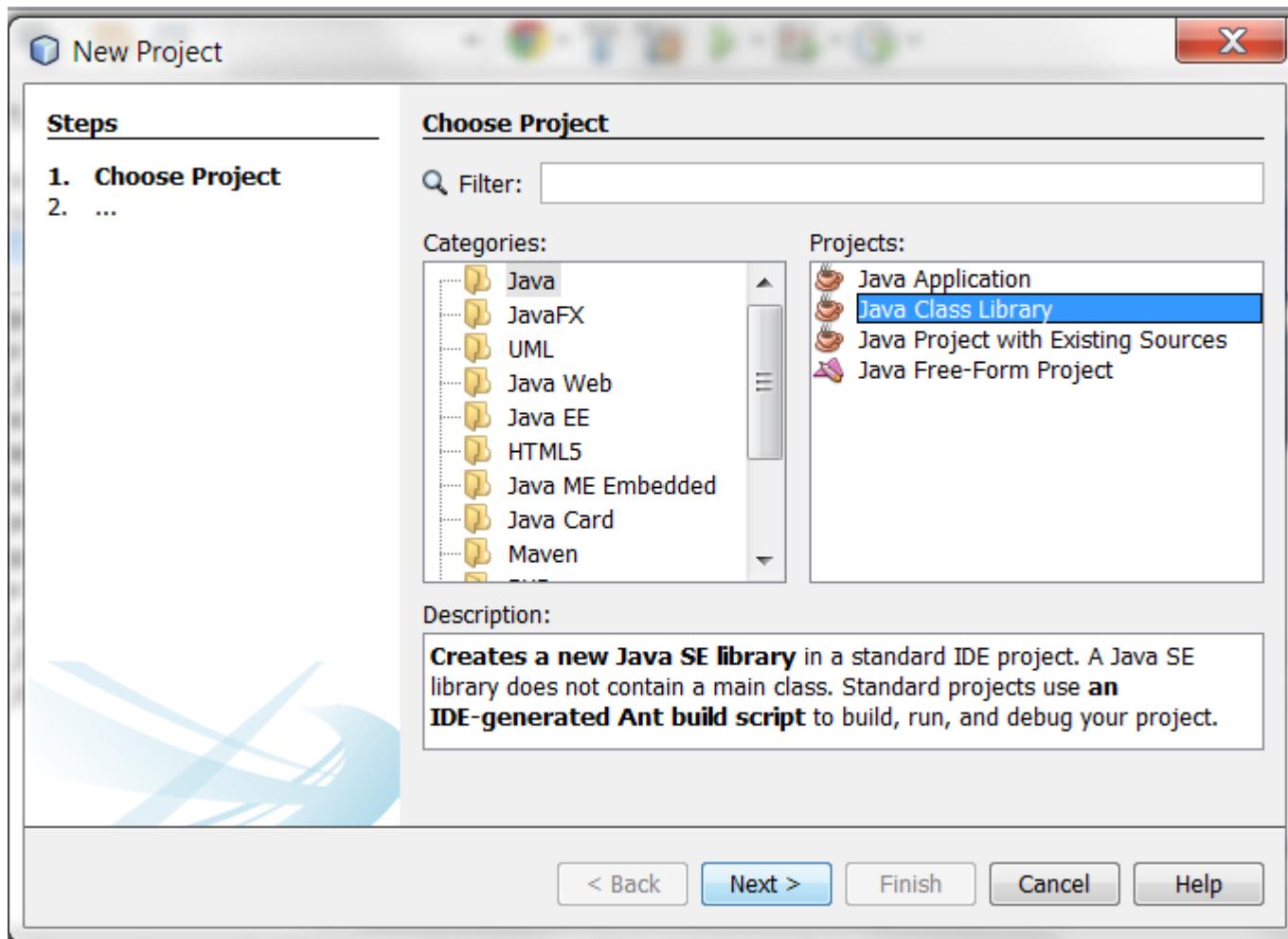


4. / 4.1./4.1.1. Creation the library project for code from lab1-2



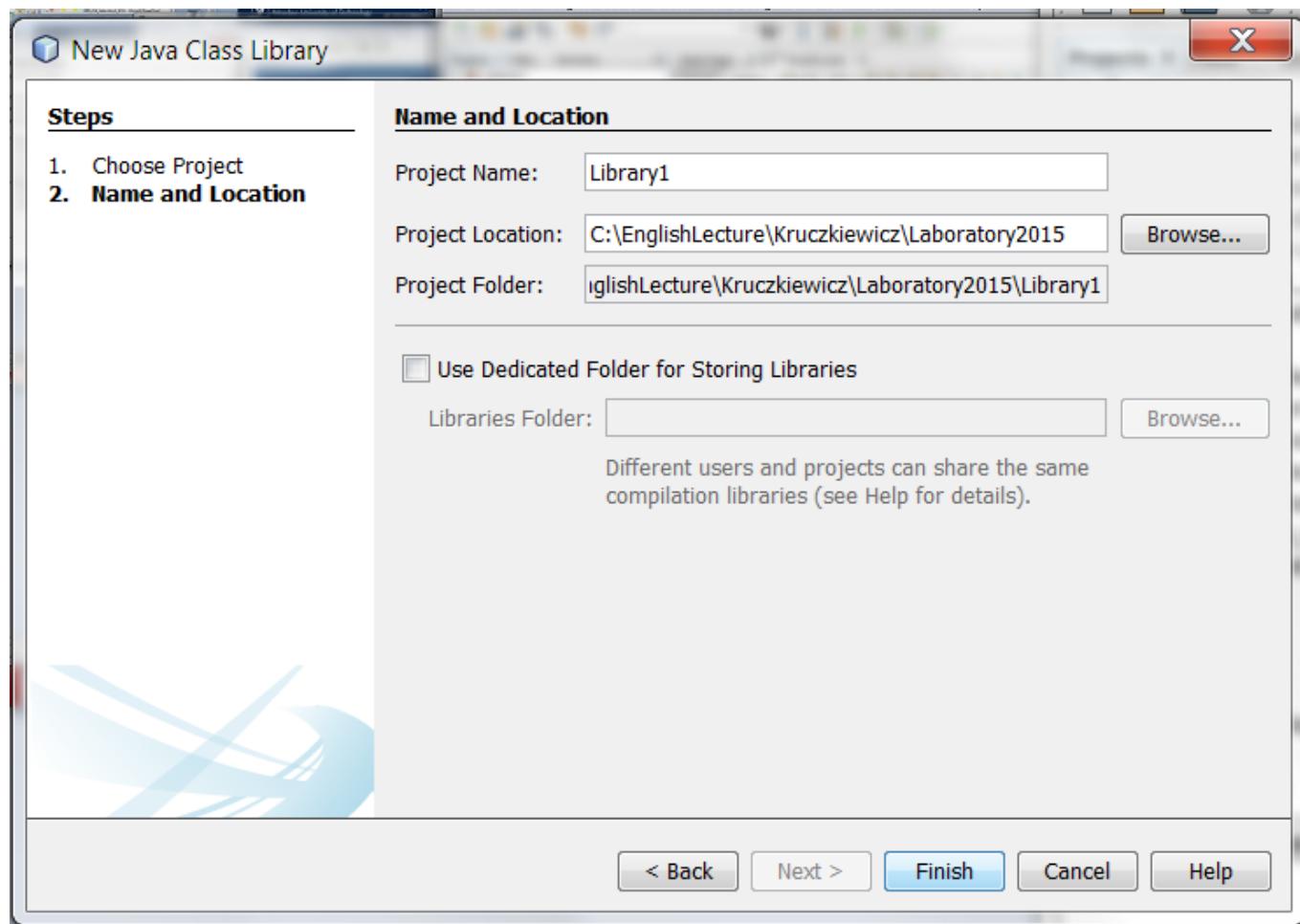


4.1.2. Creation the library project for code from lab1-2 - select Java/Java Class Library items



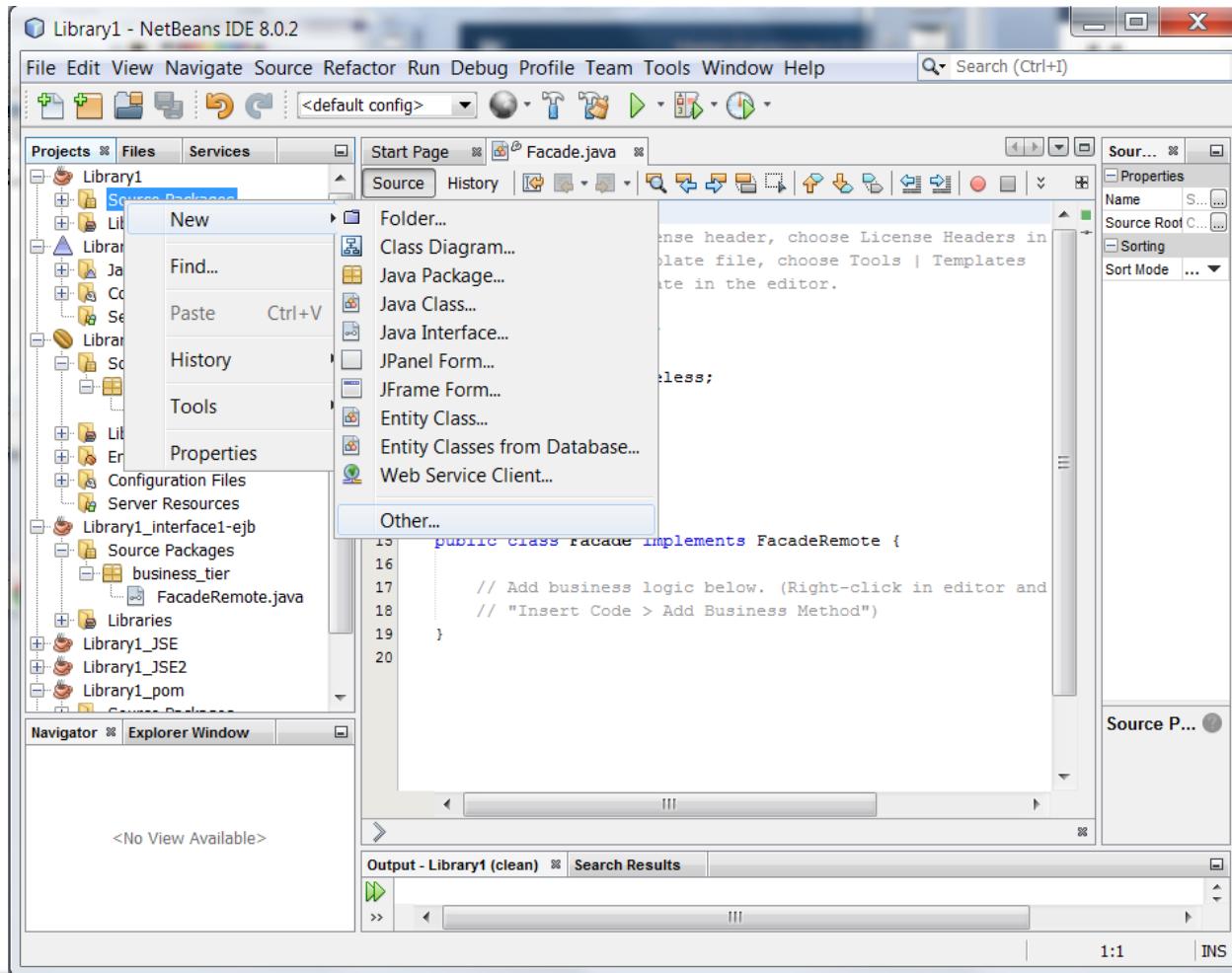


4.1.3. Creation the library project for code from lab1-2 - fill the Project Name and select the Project Location



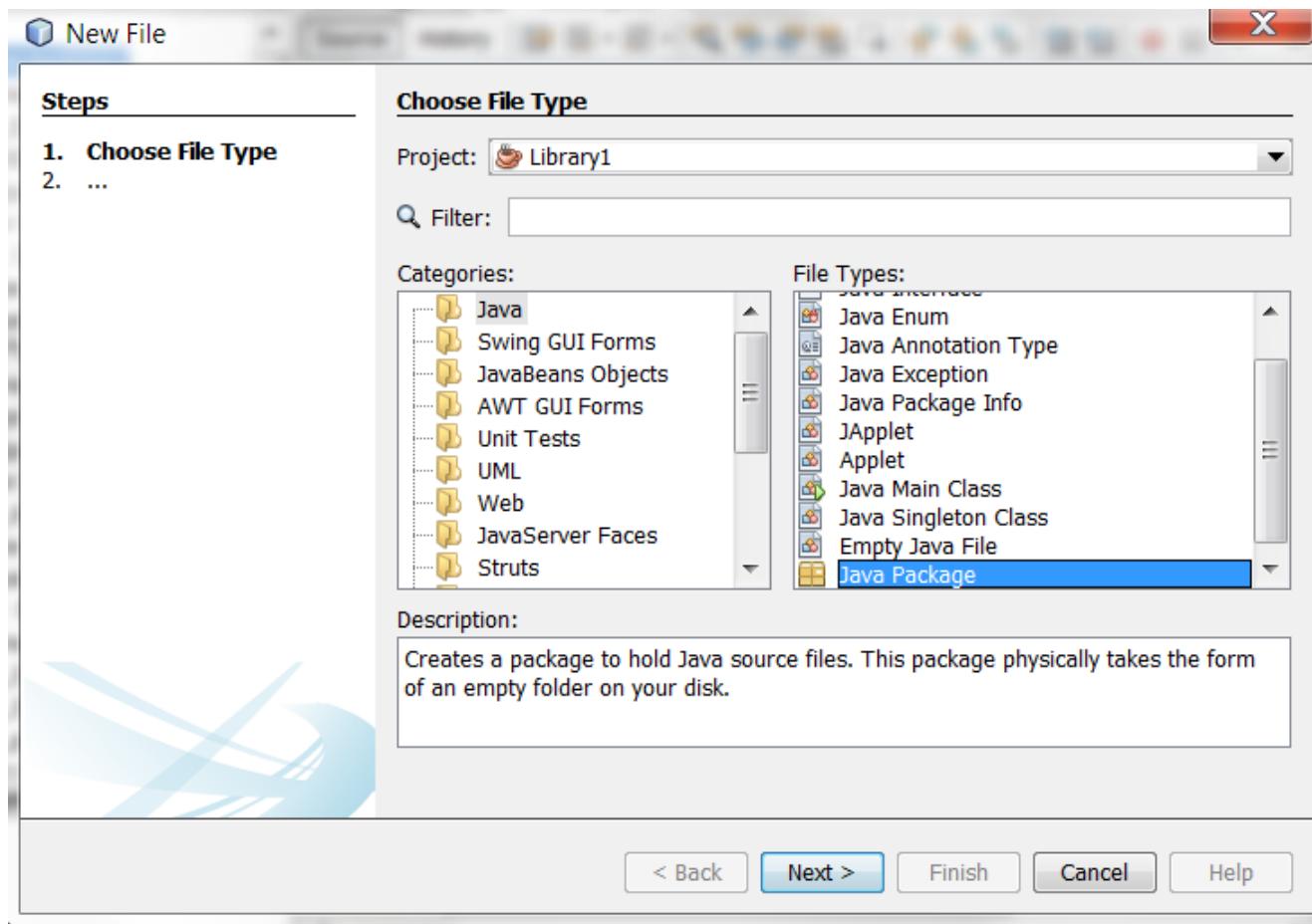


4.2. /4.2.1 Creation the package for TFacade and TFactory classes





4.2.2. Creation the package for TFacade and TFactory classes – select Java/ Java Package





4.2.3. Creation the package for TFacade and TFactory classes – fill the name of package and select location . The result on the right.

New Java Package

Steps

- Choose File Type
- Name and Location

Name and Location

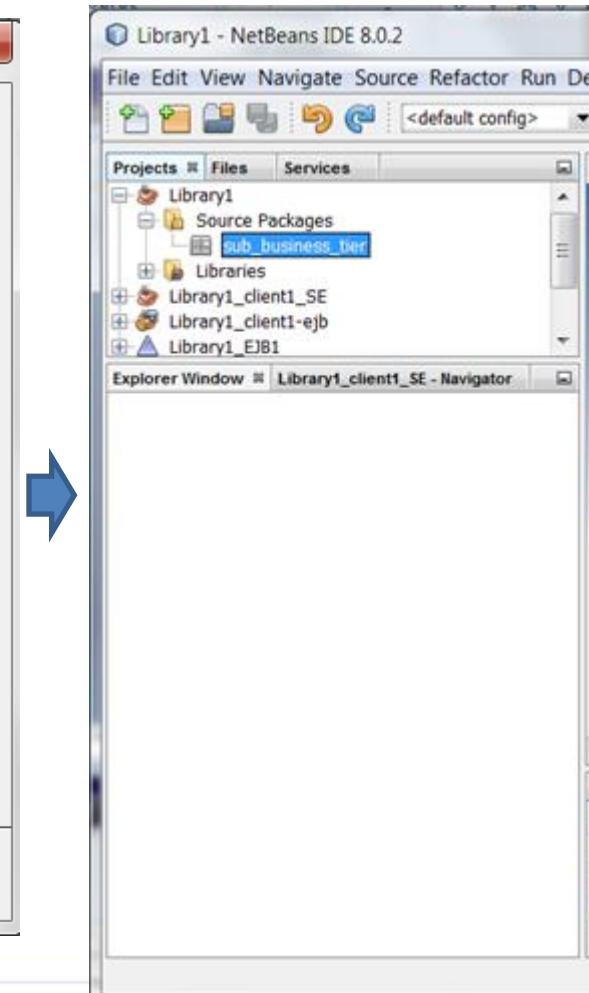
Package Name: sub_business_tier

Project: Library1

Location: Source Packages

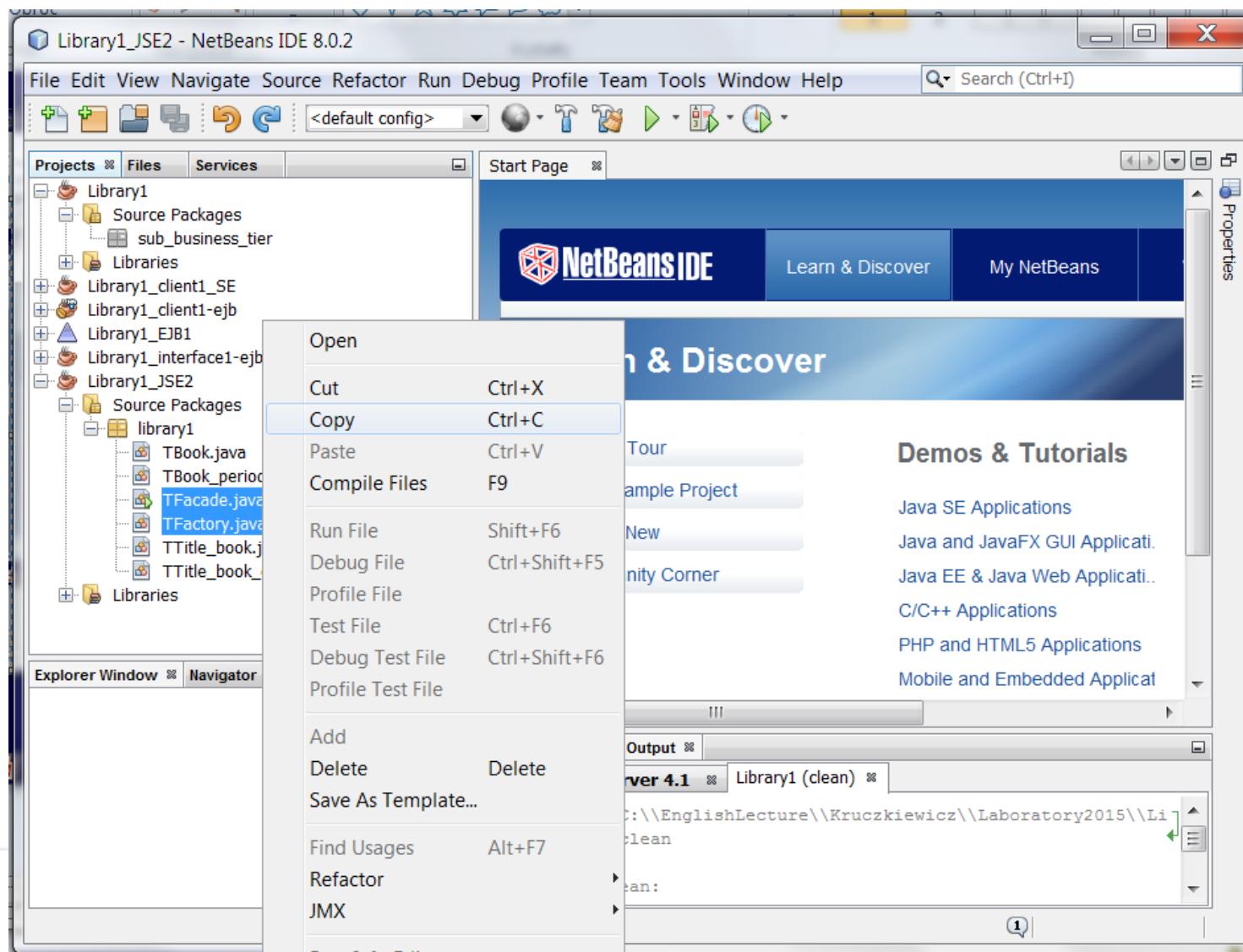
Created Folder: \Lecture\Kruczkiewicz\Laboratory2015\Library1\src\sub_business_tier

< Back Next > Finish Cancel Help



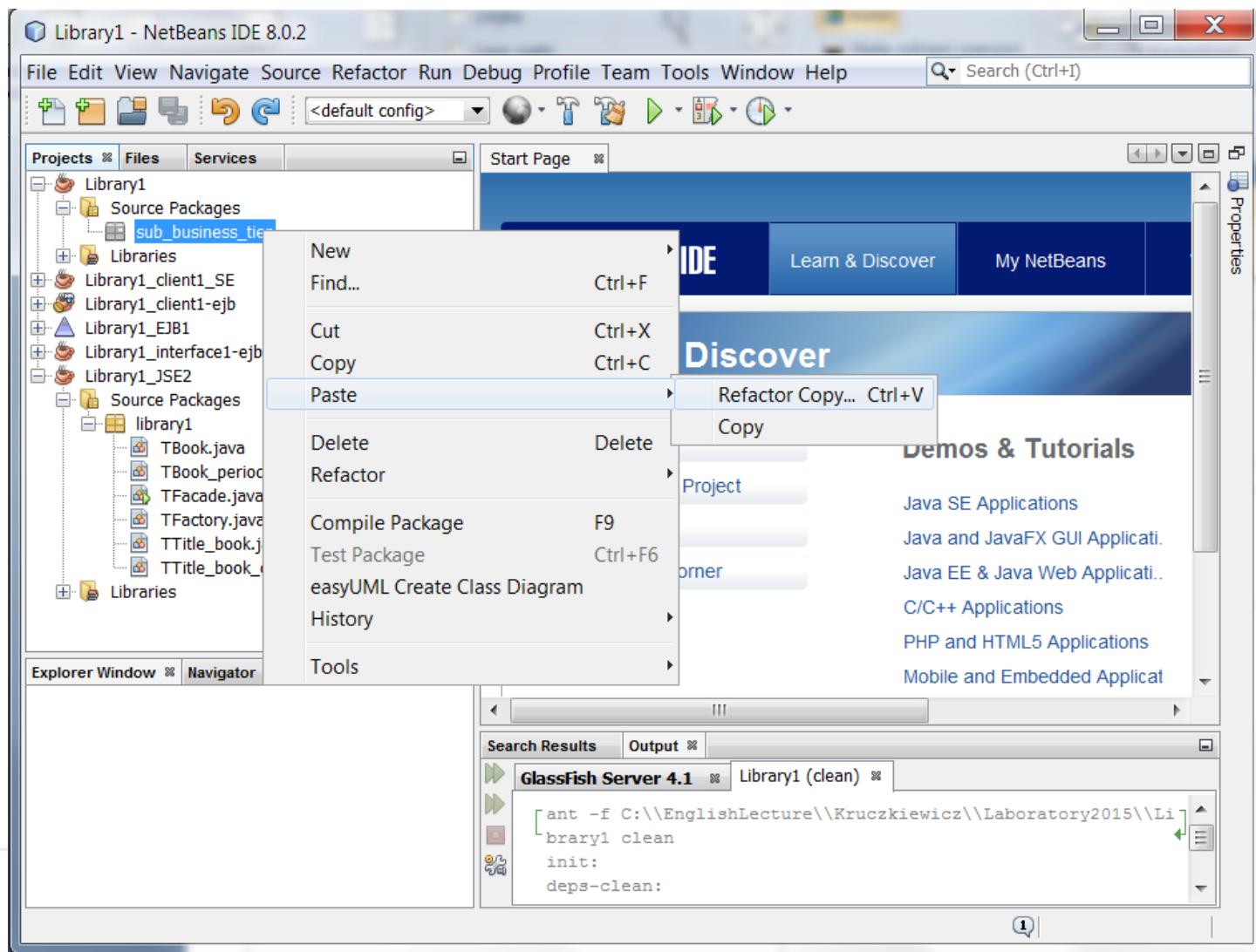


4.3. /4.3.1. Copying of TFacade and TFactory classes



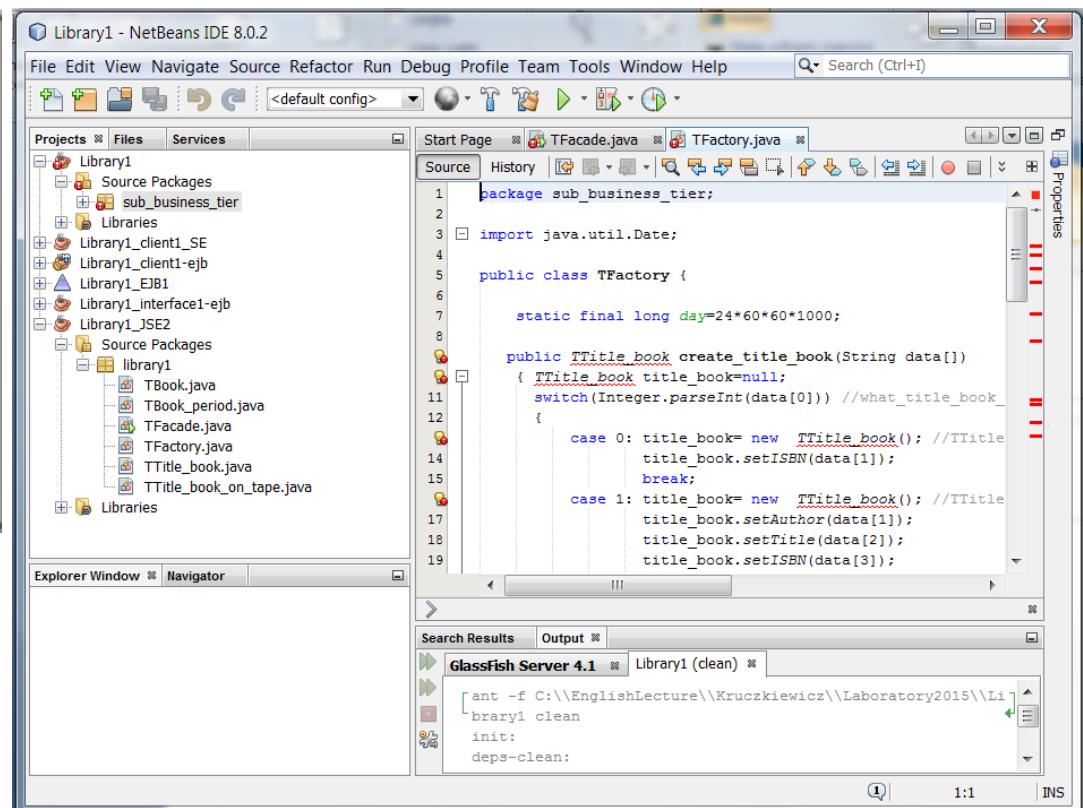
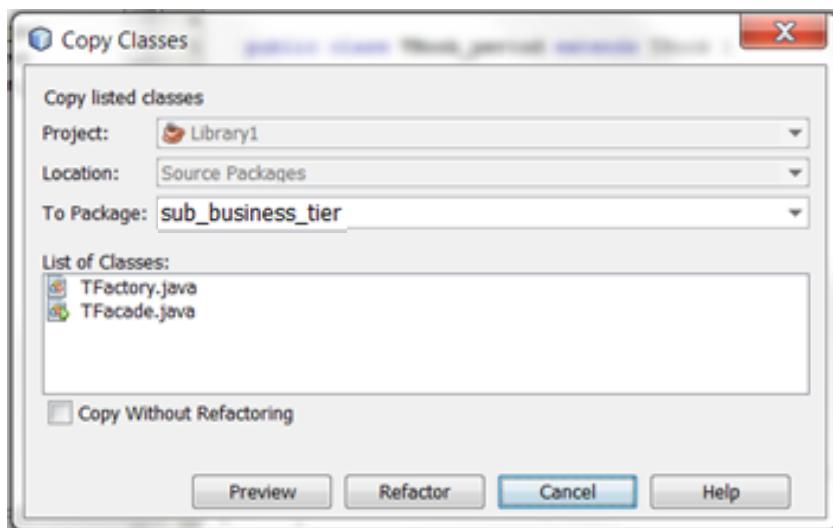


4.3.2. Paste the copies of the TFacade and TFactory classes from the lab1-2 project to the Library1 project (select the **sub_business_tier** package).





4.3.3. Paste the copies of the TFacade and TFactory classes from the lab1-2 project to the Library1 project – the refactoring process (click Refactor – on the left; the result on the right).





4.3.4. Paste the copies of the TFacade and TFactory classes from the lab1-2 project to the Library1 project – result

The screenshot shows the NetBeans IDE interface with the title bar "Library1 - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a Search field. The toolbar has icons for file operations like New, Open, Save, and Build.

The Projects tab shows the "Library1" project structure:

- Source Packages:
 - sub_business_tier
- Libraries
- Library1_client1_SE
- Library1_client1-ejb
- Library1_EJB1
- Library1_interface1-ejb
- Library1_JSE2
- Source Packages:
 - library1
 - TBook.java
 - TBook_period.java
 - TFacade.java
 - TFactory.java
 - TTITLE_book.java
 - TTITLE_book_on_tape.java
- Libraries

The Files tab shows the code editor with the "TFactory.java" file open. The code is as follows:

```
package sub_business_tier;  
  
import java.util.Date;  
  
public class TFactory {  
  
    static final long day=24*60*60*1000;  
  
    public TTITLE_book create_TITLE_book(String data[]){  
        TTITLE_book title_book=null;  
        switch(Integer.parseInt(data[0])) //what_TITLE_book_  
        {  
            case 0: title_book= new TTITLE_book(); //TTITLE  
            title_book.setISBN(data[1]);  
            break;  
            case 1: title_book= new TTITLE_book(); //TTITLE  
            title_book.setAuthor(data[1]);  
            title_book.setTitle(data[2]);  
            title_book.setISBN(data[3]);  
        }  
    }  
}
```

The Output tab shows the build logs for "GlassFish Server 4.1":

```
[ ant -f C:\\\\EnglishLecture\\\\Kruczkiewicz\\\\Laboratory2015\\\\Li  
brary1 clean  
init:  
deps-clean:
```



4.4. / 4.4.1. Creation of the new package for data classes from lab1-2

The screenshot shows the NetBeans IDE interface with the title bar "Library1 - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar has various icons for file operations like New, Open, Save, and Build. The Projects tab in the left sidebar shows a project named "Library1" with several source packages and libraries. A context menu is open over the "Source Packages" node under "Library1", with "New" selected. The submenu lists options like Folder..., Java Package..., Java Class..., Entity Classes from Database..., JSF Managed Bean..., Facelets Template Client..., Facelets Template..., Persistence Unit..., Class Diagram..., Java Interface..., JPanel Form..., JFrame Form..., Entity Class..., Web Service Client..., and Other... The main code editor window displays Java code related to book parsing and creation. The Output tab at the bottom shows a terminal window with build logs for "GlassFish Server 4.1" and "Library1 (clean)".

```
tier;  
;  
{  
    day=24*60*60*1000;  
  
    create_title_book(String data[]){  
        le_book=null;  
        parseInt(data[0])); //what_title_book_  
  
        le_book= new TTitle_book(); //TTitle  
        le_book.setISBN(data[1]);  
        ak;  
        le_book= new TTitle_book(); //TTitle  
        le_book.setAuthor(data[1]);  
        le_book.setTitle(data[2]);  
        le_book.setISBN(data[3]);  
    }  
}  
  
ant -f C:\\EnglishLecture\\Kruczkiejcz\\Laboratory2015\\Li  
brary1 clean  
init:  
deps-clean:
```



4.4.2. Creation of new package for data classes from lab1-2 – select Java/ Java Package

New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: Library1

Filter:

Categories:

- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- UML
- JMX
- Web
- JavaServer Faces

File Types:

- Java Interface
- Java Enum
- Java Annotation Type
- Java Exception
- Java Package Info
- JApplet
- Applet
- Java Main Class
- Java Singleton Class
- Empty Java File
- Java Package

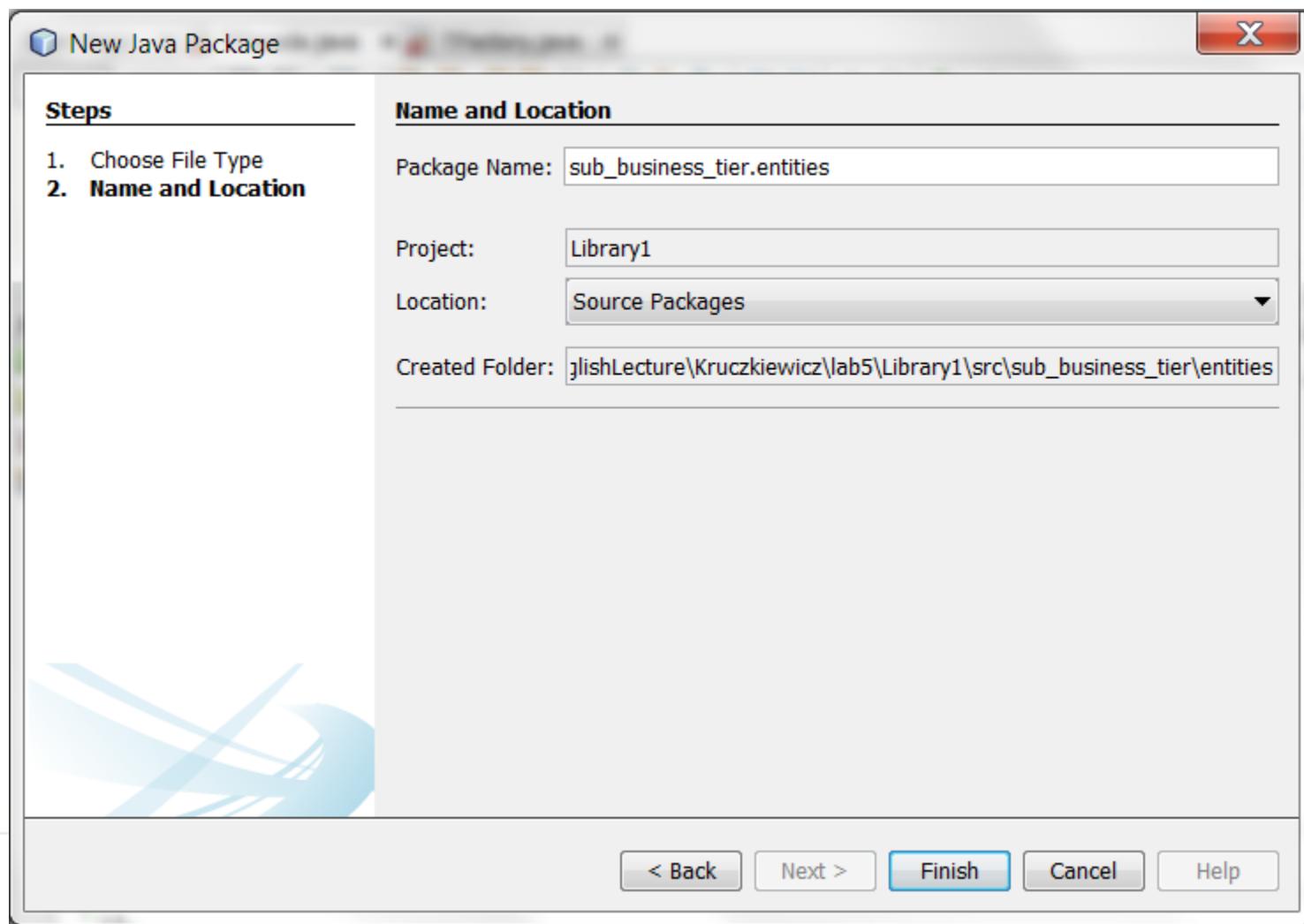
Description:

Creates a new plain Java class. This template is useful for creating new non-visual classes.

< Back Next > Finish Cancel Help



4.4.3. Creation of new package for data classes from lab1-2 – fill the name of package and select location





4.4. Result of creating of the `sub_business_tier.entities` packages

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** Library1 - NetBeans IDE 8.0.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer (Projects tab):** Shows the project structure:
 - Library1
 - Source Packages
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - Libraries
 - Library1_client1_SE
 - Library1_client1-ejb
 - Library1_EJB1
 - Library1_interface1-ejb
 - Library1_JSE2
 - Source Packages
 - library1
 - TBook.java
 - TBook_period.java
 - TFacade.java
 - TFactory.java
 - TTitle_book.java
 - TTtitle_book_on_tape.java
- Code Editor:** The current file is TFacade.java, showing the following code:

```
2 import java.util.ArrayList;
3 import java.util.Arrays;
4 import java.util.List;
5
6
7 public class TFacade {
8
9
10    // List<TTTitle_book> mTitle_books;
11    List<TTTitle_book> mTitle_books;
12
13    public TFacade() {mTitle_books = new ArrayList<>();}
14
15
16    public List<TTTitle_book> getmTitle_books() {
17        return mTitle_books;
18    }
19
20
21    void setmTitle_books(List<TTTitle_book> title_books) {
22        mTitle_books = title_books;
23    }
24}
```
- Properties Panel:** Located on the right side of the code editor.
- Output Window:** Shows the GlassFish Server 4.1 and Library1 (run) logs.
- Bottom Status Bar:** Shows the time as 6:1 and other status indicators.



4.5./4.5.1. Copying of the TTitle_book and other classes as data classes from lab1-2 project

The screenshot shows the NetBeans IDE interface with the title bar "Library1_JSE2 - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a Search field. The toolbar has various icons for file operations. The left sidebar shows the Projects view with "Source Packages" containing "sub_business_tier" with files TFacade.java and TFactory.java, and "sub_business_tier.entities". It also shows "Libraries" with "Library1_client1_SE", "Library1_client1-ejb", "Library1_EJB1", "Library1_interface1-ejb", and "Library1_JSE2". The "Library1_JSE2" node contains "Source Packages" with "library1" containing "TBook.java", "TBook_period.java", "TFacade.java", "TFactory.java", "TTitle_book.java", and "TTitle_book_on_tape.java". The "Libraries" section lists "Apache Commons Lang 3.3.0", "Java API for XML Processing 2.6.2", "Java API for XML Processing 2.6.2 API", "Java API for XML Processing 2.6.2 Javadoc", "Java API for XML Processing 2.6.2 Source", and "JBoss Seam Core 3.0.0.Final". The main editor area shows the "TFacade.java" file with the following code:

```
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
```

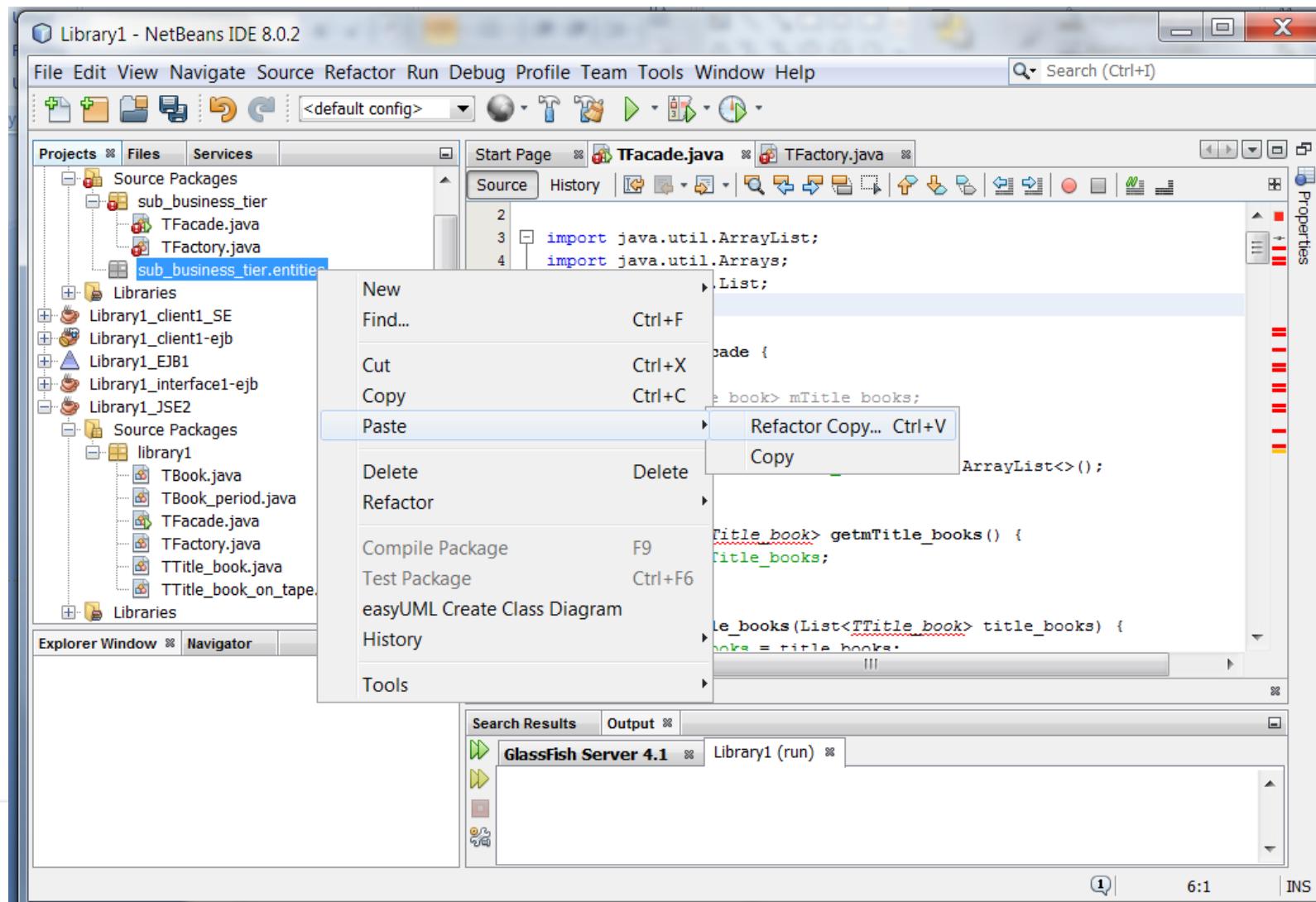
A context menu is open over the code, with "Copy" highlighted. Other options in the menu include Open, Cut (Ctrl+X), Paste (Ctrl+V), Compile Files (F9), Run File (Shift+F6), Debug File (Ctrl+Shift+F5), Profile File, Test File (Ctrl+F6), Debug Test File (Ctrl+Shift+F6), Profile Test File, Add, Delete, Save As Template..., Find Usages (Alt+F7), Refactor, and JMX.

The code editor shows the following Java code:

```
'acade {  
    @XmlElementWrapper(name = "title_books")  
    @XmlElement(name = "book")  
    private List<TTitle_book> mTitle_books;  
  
    public void setmTitle_books(List<TTitle_book> mTitle_books) {  
        this.mTitle_books = mTitle_books;  
    }  
  
    public List<TTitle_book> getmTitle_books() {  
        return mTitle_books;  
    }  
  
    public void addTitle_books(TTitle_book title_books) {  
        mTitle_books.add(title_books);  
    }  
}
```

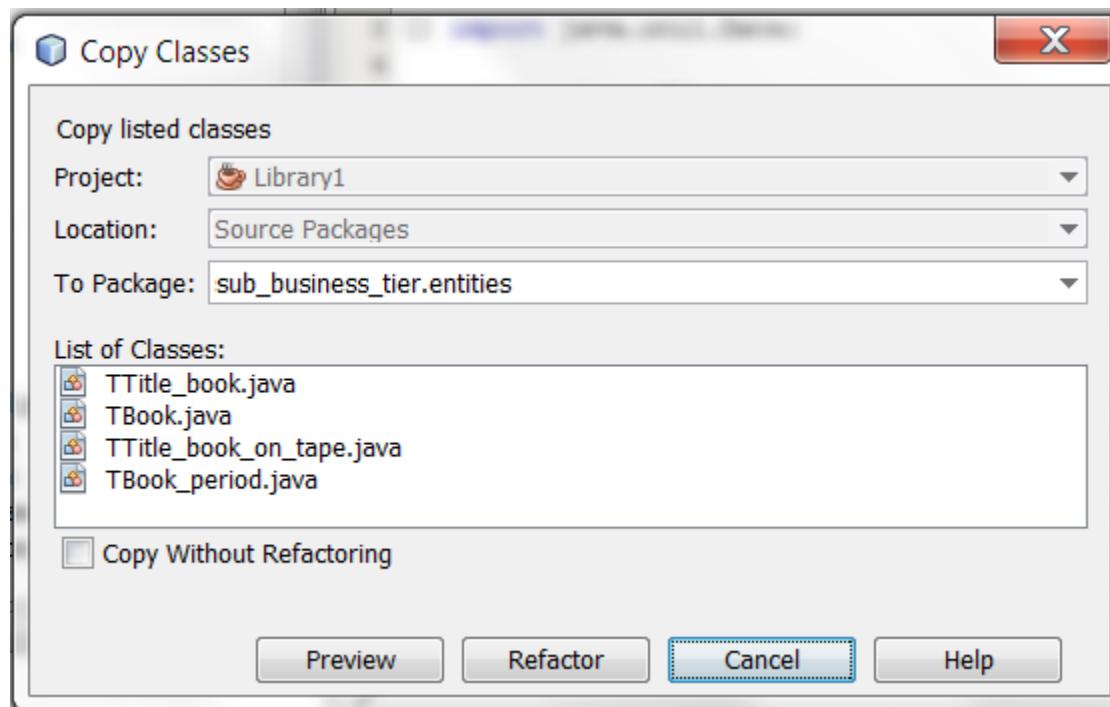


4.5.2 Paste the copies of the TTtitle_book and other classes as data classes from the lab1-2 project to the Library1 project (select the **sub_business_tier.entities** package).





4.5.3. Paste the copies of the TTitle_book and other classes as data classes from the lab1-2 project to the Library1 project – the refactoring process (click the Refactor button).





4.5.4. Paste the copies of the TTitle_book and other classes as data classes from the lab1-2 project to the Library1 project - result

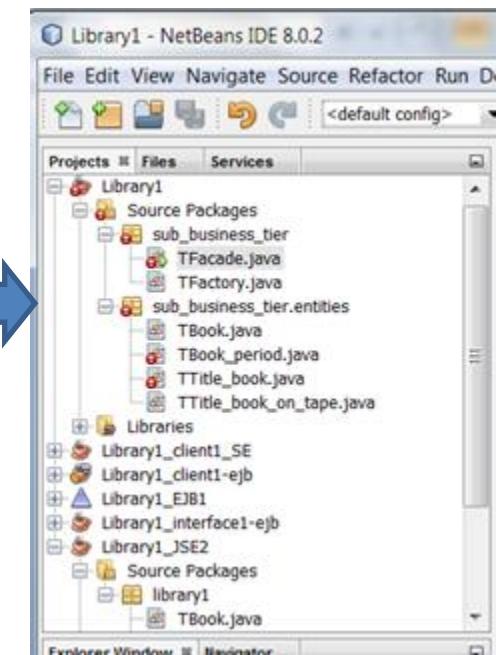
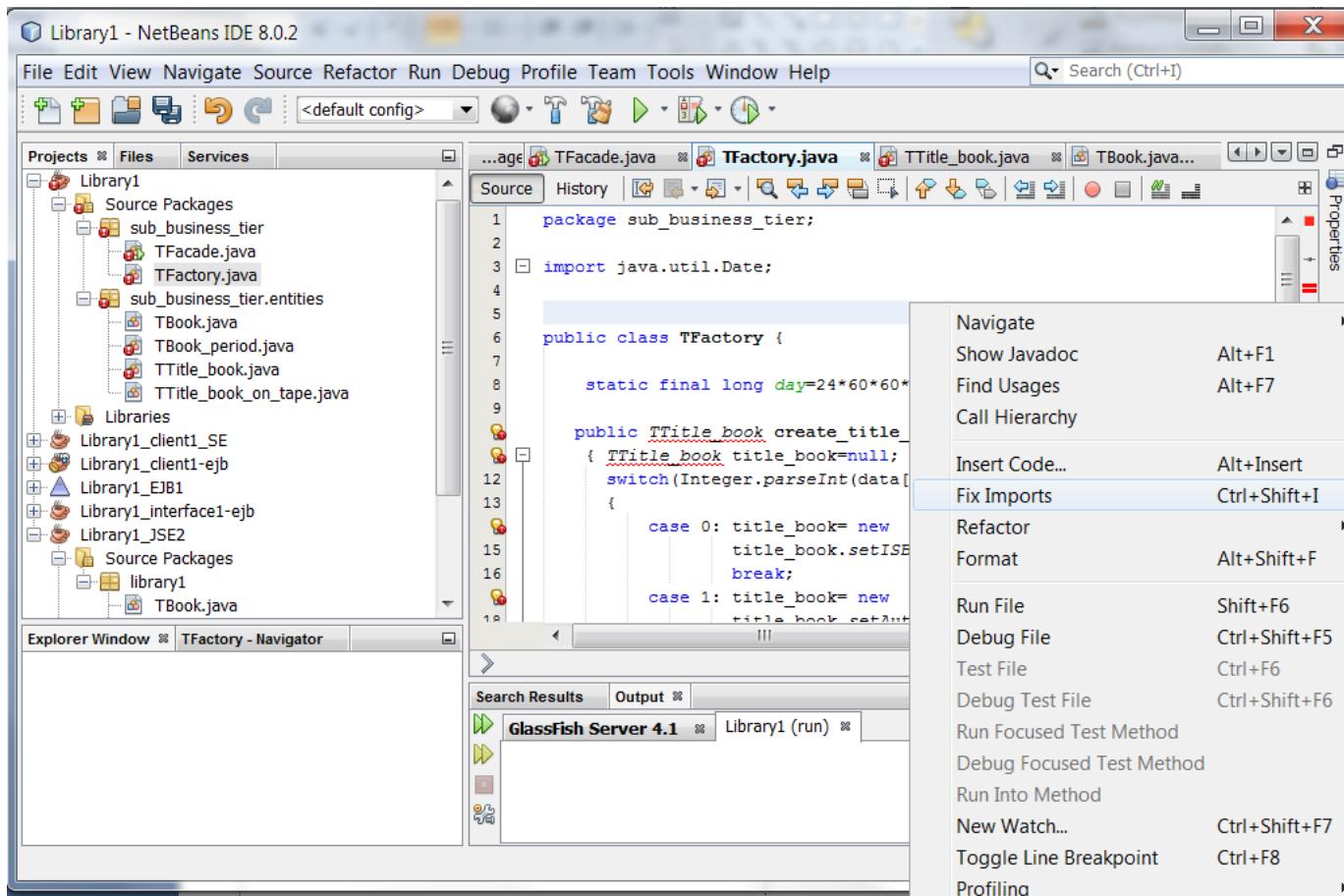
The screenshot shows the NetBeans IDE interface with the title bar "Library1 - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a Search field. The toolbar has icons for file operations like New, Open, Save, and Build.

The Projects tab shows the "Library1" project structure:

- Source Packages:
 - sub_business_tier
 - TFacade.java
 - TFactory.java
 - sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java
- Libraries:
 - Library1_client1_SE
 - Library1_client1-ejb
 - Library1_EJB1
 - Library1_interface1-ejb
 - Library1_JSE2
 - Source Packages
 - library1
 - TBook.java



4.6. /4.6.1. „Fix Imports” process for adding the lacking imports of classes – the TFactory class





4.6.2. „Fix Imports” process for adding the lacking imports of classes – the TFacade class

```
1 package sub_business_tier;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7
8 public class TFacade {
9
10    // List<TTitle_book> mTitle_books;
11    List<TTitle_book> mTitle_books;
12
13    public TFacade() {mTitle_books = new ArrayList();}
14
15    public List<TTitle_book> getmTitle_books() {
16        return mTitle_books;
17    }
18}
```

```
1 package sub_business_tier;
2
3 import java.util.ArrayList;
4 import java.util.Arrays;
5 import java.util.List;
6
7
8 public class TFacade {
9
10    // List<TTitle_book> mTitle_books;
11    List<TTitle_book> mTitle_books;
12
13    public TFacade() {mTitle_books = new ArrayList();}
14
15    public List<TTitle_book> getmTitle_books() {
16        return mTitle_books;
17    }
18}
```





4.6.3. „Fix Imports” process for adding the lacking imports of classes – the TTitle_book class

Library1 - NetBeans IDE 8.0.2

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

Search (Ctrl+I)

Projects Files Services

Source Packages

- sub_business_tier
 - TFacade.java
 - TFactory.java
- sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java

Libraries

- Library1_client1_SE
- Library1_client1-ejb
- Library1_EJB1
- Library1_interface1-ejb
- Library1_JSE2
- Source Packages
 - library1
 - TBook.java

Explorer Window equals - Navigator

141 if (getISBN().equals(((TTitle_book) obj).getISBN())) {
142 if (getActor().equals(((TTitle_book) obj).getActor())) {
143 result = true;
144 }
145 }
146 return result;

149 public ArrayList<String> add_L
TFactory factory = new TFa
TBook newbook;
newbook = factory.create_L
if (search_book(newbook) =
mBooks.add(newbook);
newbook.setmTitle_book
return getbooks();
}
return null;

sub_business_tier.entities.TTitle_book > equals

Search Results Output

GlassFish Server 4.1 Library1 (run)

Context menu open over the code at line 149:

- 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

Library1 - NetBeans IDE 8.0.2

File Edit View Navigate Source Refactor Run

Search (Ctrl+I)

Projects Files Services

Source Packages

- sub_business_tier
 - TFacade.java
 - TFactory.java
- sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTitle_book_on_tape.java

Libraries

- Library1_client1_SE
- Library1_client1-ejb
- Library1_EJB1
- Library1_interface1-ejb
- Library1_JSE2
- Source Packages
 - library1
 - TBook.java

Explorer Window startPeriod - Navigator





4.6.4. „Fix Imports” process for adding the lacking imports of classes – the TBook_period class

The screenshot shows the NetBeans IDE interface with the title bar "Library1 - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New Project, Open Project, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The Projects tab is selected in the left sidebar, showing a hierarchy of packages and files. The main editor window displays Java code for the TBook_period.java class. A context menu is open over the code, with the "Fix Imports" option highlighted. The menu also includes options like Navigate, Insert Code..., Refactor, Format, Run File, Debug File, Test File, Debug Test File, Run Focused Test Method, Debug Focused Test Method, Run Into Method, New Watch..., and Toggle Line Breakpoint. The keyboard shortcut for Fix Imports is Ctrl+Shift+I. The bottom of the screen shows the GlassFish Server 4.1 and Library1 (run) toolbars.

This screenshot shows the same NetBeans IDE interface after the "Fix Imports" process has been completed. The code in the TBook_period.java editor now includes the necessary import statement for the TFactory class. The context menu is no longer open, and the code editor shows the updated imports. The rest of the interface, including the Projects tab and the GlassFish Server 4.1 toolbar, remains the same.





4.6.5. The result

Library1 - NetBeans IDE 8.0.2

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

Search (Ctrl+I)

Projects Files Services

Source History

Properties

...ava TBook.java TTtitle_book_on_tape.java TBook_period.java

```
public void setPeriod(Date date) {
    period = date;
}

@Override
public boolean period_pass(Object data) {
    Date date = TFactory.mdays((String) data);
    return period.compareTo(date) < 0;
}

@Override
public void startPeriod(Object data)
{
    Date help = TFactory.mdays((String) data);
    setPeriod(help);
}

@Override
public String toString() // your code here
```

sub_business tier.entities.TBook_period > period_pass >

Explorer Window period_pass - Navigator

Search Results Output

GlassFish Server 4.1 Library1 (run)

TBook_period.java saved.

20:14 INS



4.7. Build and Run the Library1 Project

Library1 - NetBeans IDE 8.0.2

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

Search (Ctrl+I)

Projects Files Services

Source Packages

- sub_business_tier
 - TFacade.java
 - TFactory.java
- sub_business_tier.entities
 - TBook.java
 - TBook_period.java
 - TTitle_book.java
 - TTtitle_book_on_tape.java

Libraries

- Library1_client1_SE
- Library1_client1-ejb
- Library1_EJB1
- Library1_interface1-ejb
- Library1_JSE2

Source Packages

- library1
 - TBook.java
 - TBook_period.java
 - TFacade.java

Output

GlassFish Server 4.1 Library1 (run) *

```
run:
[
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2,
Title: Title3 Author: Author3 ISBN: ISBN3 Publisher: Publisher3,
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Actor: Actor2,
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4]
[
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Number: 1][
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1][
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 1,
Title: Title2 Author: Author2 ISBN: ISBN2 Publisher: Publisher2 Number: 2][
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 3 Period: Sat Mar 14 18:52:46 CET 2015][
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 3 Period: Sat Mar 14 18:52:46 CET 2015,
Title: Title1 Author: Author1 ISBN: ISBN1 Publisher: Publisher1 Actor: Actor1 Number: 2 Period: Tue Mar 10 18:52:46 CET 2015][
Title: Title4 Author: Author4 ISBN: ISBN4 Publisher: Publisher4 Actor: Actor4 Number: 2]

Titles of book
[Publisher1, ISBN1, Title1, Author1, ]
[Publisher2, ISBN2, Title2, Author2, ]
[Publisher3, ISBN3, Title3, Author3, ]
[Publisher1, ISBN1, Title1, Author1, Actor1]
[Publisher2, ISBN2, Title2, Author2, Actor2]
[Publisher4, ISBN4, Title4, Author4, Actor4]
```

Members

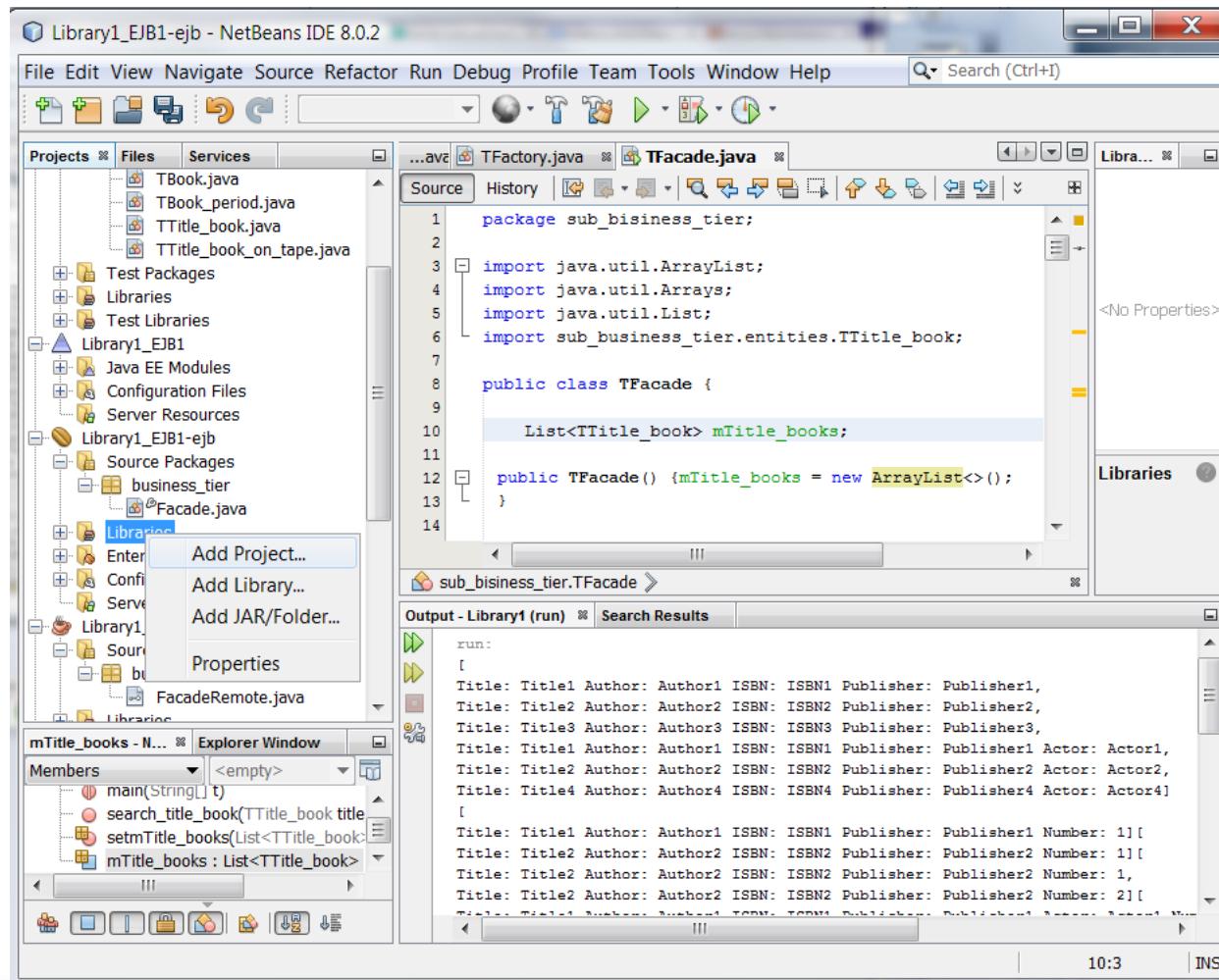
- TBook_period :: TBook
 - getPeriod() : Date
 - period_pass(Object data) : boolean
 - setPeriod(Date date)
 - startPeriod(Object data)
 - toString() : String
- period : Date

Navigation icons

20:14 INS

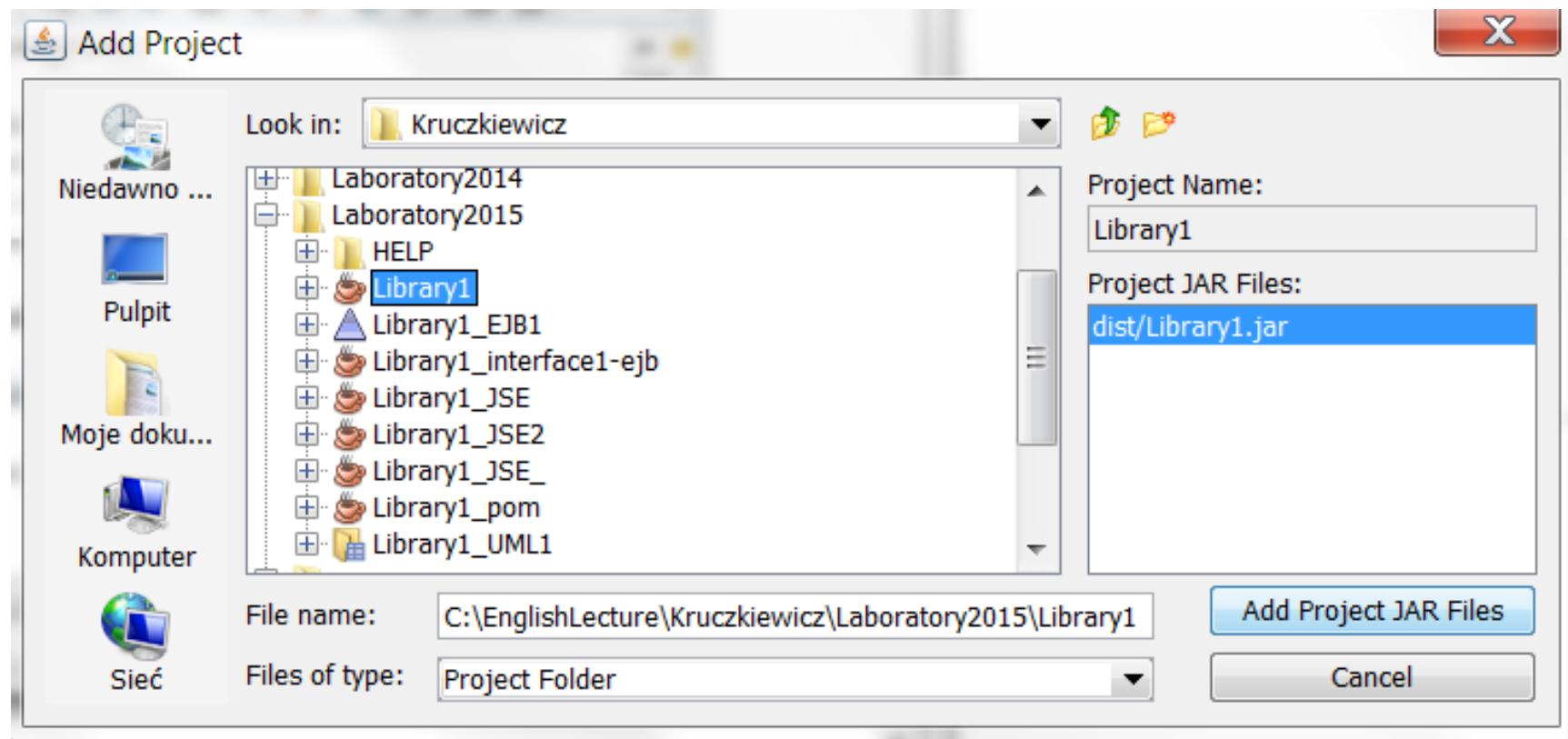


5./5.1./5.1.1. Adding the library project (p.4) to the EJB module of EE project



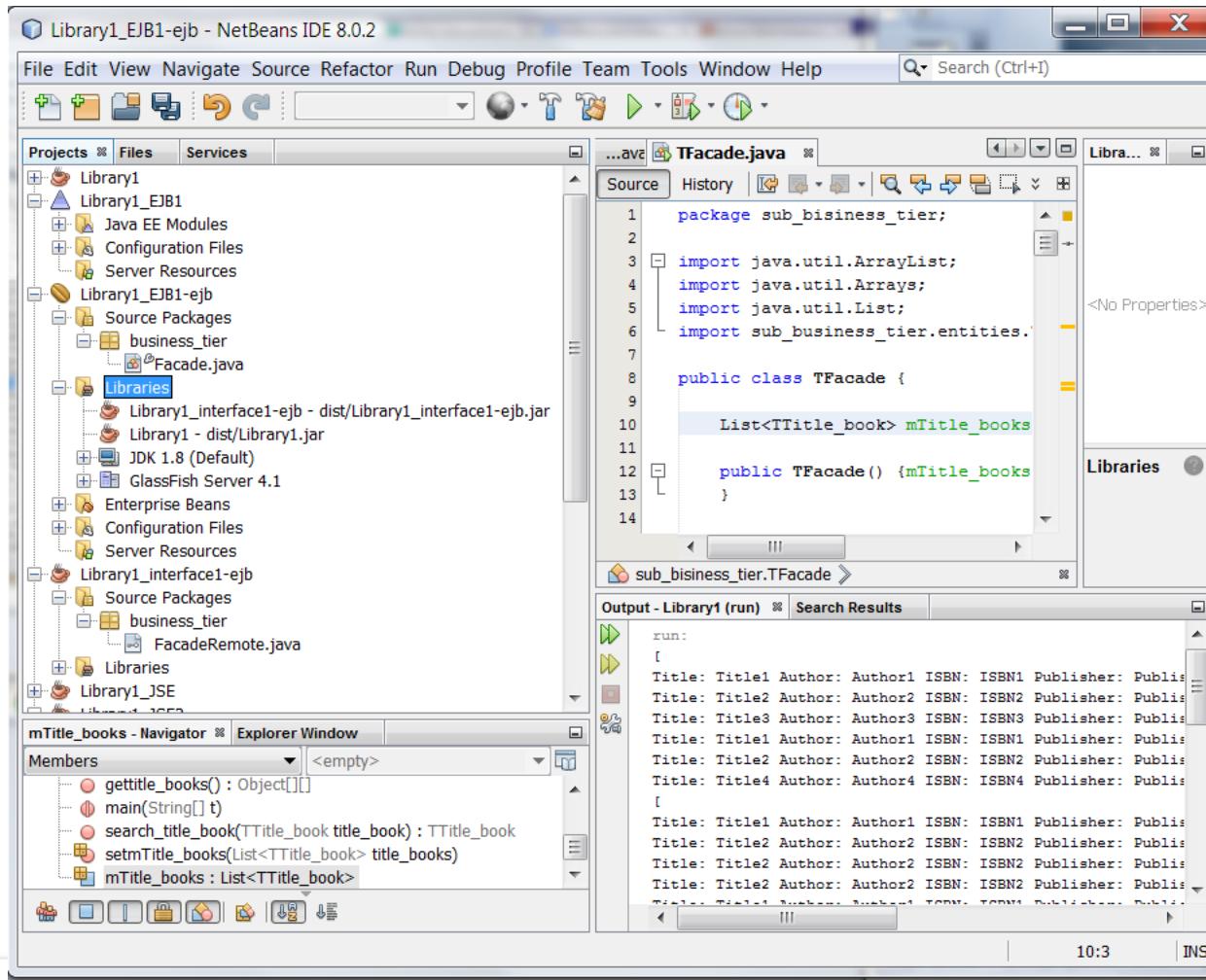


5.1.2. Adding the library project (p.4) to the EJB module of EE project - select the Library1 project and click the Add Project JAR Files





5.1.3. Adding the library project (p.4) to the EJB module of EE project p select the Library1 project - result





5.2. /5.2.1. Definition of interface of EJB Facade (as the adapter of methods of POJO TFacade)

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** Library1_interface1-ejb - NetBeans IDE 8.0.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar.
- Projects Tab:** Shows the project structure:
 - Library1
 - Library1_EJB1
 - Java EE Modules
 - Configuration Files
 - Server Resources
 - Library1_EJB1-ejb
 - Source Packages
 - business_tier
 - Facade.java
 - Libraries
 - Library1_interface1-ejb - dist/Library1_interface1-ejb.jar
 - Library1 - dist/Library1.jar
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
 - Enterprise Beans
 - Configuration Files
 - Server Resources
 - Library1_interface1-ejb
 - Source Packages
 - business_tier
 - FacadeRemote.java
 - Libraries
- Code Editor:** Displays the `FacadeRemote.java` file content:

```
5  /*
6  package business_tier;
7
8  import javax.ejb.Remote;
9
10 /**
11 *
12 * @author Zofia
13 */
14
15 @Remote
16 public interface FacadeRemote {
17     public Object[][] gettitle_books();
18
19     public String add_title_book(String data[]);
20
21     public ArrayList<String> add_book(String data[], String data2[]);
22
23     public ArrayList<String> Search_title_book(String data[]);
24
25     public String Search_accessible_book(String data1[], Object data2);
26 }
```
- Navigator Tab:** Shows the members of the `FacadeRemote` interface:
 - Search_accessible_book(String[] data1, Object data2) : String
 - Search_title_book(String[] data) : <any>
 - add_book(String[] data1, String[] data2) : <any>
 - add_title_book(String[] data) : String
 - gettitle_books() : Object[][]
- Output Tab:** Shows the output for the run configuration "Library1 (run)".





5.2.2. „Fix Imports” process

A screenshot of an IDE interface showing a Java code editor. The code is for a FacadeRemote interface. A context menu is open at the cursor position, with 'Fix Imports' highlighted. Other options in the menu include 'Navigate', 'Show Javadoc', 'Find Usages', 'Call Hierarchy', 'Insert Code...', 'Refactor', 'Format', 'Run File', 'Debug File', 'Test File', 'Debug Test File', 'Run Focused Test Method', 'Debug Focused Test Method', 'Run Into Method', 'New Watch...', 'Toggle Line Breakpoint', 'Profiling', and 'Cut'. The code in the editor includes imports for javax.ejb.Remote and java.util.ArrayList.

```
5  /*
6  package business_tier;
7
8  import javax.ejb.Remote;
9
10 /**
11 *
12 * @author Zofia
13 */
14 @Remote
15 public interface FacadeRemote {
16     public Object[][] gettitle_books();
17
18     public String add_title_book(String data[]);
19
20     public ArrayList<String> add_book(String
21
22     public ArrayList<String> Search_title_boo
23
24     public String Search_accessible_book(Stri
25
26 }
```

5.2.3. „Fix Imports” process - result

A screenshot of the same IDE interface after the 'Fix Imports' process has been run. The code now includes the missing import statement for javax.ejb.Remote. The code remains identical to the previous screenshot, except for the addition of the import statement at line 9.

```
6  package business_tier;
7
8  import java.util.ArrayList;
9  import javax.ejb.Remote;
10
11 /**
12 * @author Zofia
13 */
14 @Remote
15 public interface FacadeRemote {
16
17     public Object[][] gettitle_books();
18
19     public String add_title_book(String data[]);
20
21     public ArrayList<String> add_book(String data1[], String data2[]);
22
23     public ArrayList<String> Search_title_book(String data[]);
24
25     public String Search_accessible_book(String data1[], Object data2);
26 }
```





5.2.4 – Definition of the FacadeRemote interface

```
package business_tier;  
  
import java.util.ArrayList;  
import javax.ejb.Remote;  
  
@Remote  
public interface FacadeRemote {  
  
    public Object[][] gettitle_books();  
  
    public String add_title_book(String data[]);  
  
    public ArrayList<String> add_book(String data1[], String data2[]);  
  
    public ArrayList<String> Search_title_book(String data[]);  
  
    public String Search_accessible_book(String data1[], Object data2);  
}
```





5.3. /5.3.1. Definition of EJB Facade as the Session Bean type attribute in the Library_EJB1 – ejb module (as the adapter of methods of the POJO TFacade class) – before Fix Imports

The screenshot shows the NetBeans IDE interface with the title bar "Library1_EJB1-ejb - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a Search field. The toolbar has icons for file operations like New, Open, Save, and Build.

The Projects tab shows the project structure:

- Library1
- Source Packages
 - sub_business_tier
 - sub_business_tier.entities
- Test Packages
- Libraries
- Test Libraries
- Library1_EJB1
 - Java EE Modules
 - Configuration Files
 - Server Resources
- Library1_EJB1-ejb
 - Source Packages
 - business_tier
 - Facade.java
 - Libraries
 - Library1_interface1-ejb - dist/Library1_interface1-ejb.jar
 - Library1 - dist/Library1.jar
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
- Enterprise Beans
- Configuration Files
- Server Resources
- Library1_interface1-ejb
 - Source Packages
 - business_tier
 - FacadeRemote.java
 - Libraries
 - Java EE 6 API Library - javaee-api-6.0.jar
 - JDK 1.8 (Default)

The Facade.java code editor displays the following Java code:

```
...5 lines
package business_tier;

import javax.ejb.Stateless;

@Stateless
public class Facade implements FacadeRemote {
    TFacade facade = new TFacade();
    @Override
    public Object[][] gettitle_books() {
        return facade.gettitle_books();
    }
    @Override
    public String add_title_book(String data[]) {
        return facade.add_title_book(data);
    }
    @Override
    public ArrayList<String> add_book(String data1[], String data2[]) {
        return facade.add_book(data1, data2);
    }
    @Override
    public ArrayList<String> Search_title_book(String data[]) {
        return facade.Search_title_book(data);
    }
    @Override
    public String Search_accessible_book(String data1[], Object data2) {
        return facade.Search_accessible_book(data1, data2);
    }
}
```

The Navigator and Explorer Window panes are visible at the bottom, and the Output and Search Results panes are shown at the bottom right. The status bar at the bottom right shows the time as 8:28 and the mode as INS.



5.3.2. Definition of EJB Facade as the Session Bean type attribute in the Library_EJB1 – ejb module (as the adapter of methods of the POJO TFacade class) - after Fix Imports

The screenshot shows the NetBeans IDE interface with the title bar "Library1_EJB1-ejb - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a search field "Search (Ctrl+I)". The toolbar has icons for file operations like New, Open, Save, and Run.

The left pane displays the Project Explorer with the following structure:

- Library1
- Source Packages
 - sub_business_tier
 - sub_business_tier.entities
- Test Packages
- Libraries
- Test Libraries
- Library1_EJB1
 - Java EE Modules
 - Configuration Files
 - Server Resources
- Library1_EJB1-ejb
 - Source Packages
 - business_tier
 - Facade.java
 - Libraries
 - Library1_interface1-ejb - dist/Library1_interface1-ejb.jar
 - Library1 - dist/Library1.jar
 - JDK 1.8 (Default)
 - GlassFish Server 4.1
- Enterprise Beans
- Configuration Files
- Server Resources
- Library1_interface1-ejb
 - Source Packages
 - business_tier
 - FacadeRemote.java
 - Libraries
 - Java EE 6 API Library - javaee-api-6.0.jar
 - JDK 1.8 (Default)

```
...ge Facade.java ... TBook.java ... TTitle_book_on_tape.java ... TTitle_book.java...
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)
Source History ...
1  ...5 lines
6 package business_tier;
7
8 import java.util.ArrayList;
9 import javax.ejb.Stateless;
10 import sub_bisness_tier.TFacade;
11
12 @Stateless
13 public class Facade implements FacadeRemote {
14     TFacade facade = new TFacade();
15
16     @Override
17     public Object[][] gettitle_books() {
18         return facade.gettitle_books();
19     }
20
21     @Override
22     public String add_title_book(String data[]) {
23         return facade.add_title_book(data);
24     }
25
26     @Override
27     public ArrayList<String> add_book(String data1[], String data2[]) {
28         return facade.add_book(data1, data2);
29     }
30
31 }
```

Facade - Navigator > Explorer Window

Members

- Facade :: FacadeRemote
- Search_accessible_book(String[] data1, Object data2) : String

Output - Library1 (run) > Search Results



```
package business_tier;

import java.util.ArrayList;
import javax.ejb.Stateless;
import sub_business_tier.TFacade;
import sub_business_tier.entities.TBook;
import sub_business_tier.entities.TTitle_book;

@Stateless
public class Facade implements FacadeRemote {

    TFacade facade = new TFacade();

    @Override
    public Object[][] gettitle_books() {
        return facade.gettitle_books();  }

    @Override
    public String add_title_book(String data[]) {
        return facade.add_title_book(data);  }

    @Override
    public ArrayList<String> add_book(String data1[], String data2[]) {
        return facade.add_book(data1, data2);  }

    @Override
    public ArrayList<String> Search_title_book(String data[]) {
        return facade.Search_title_book(data);  }

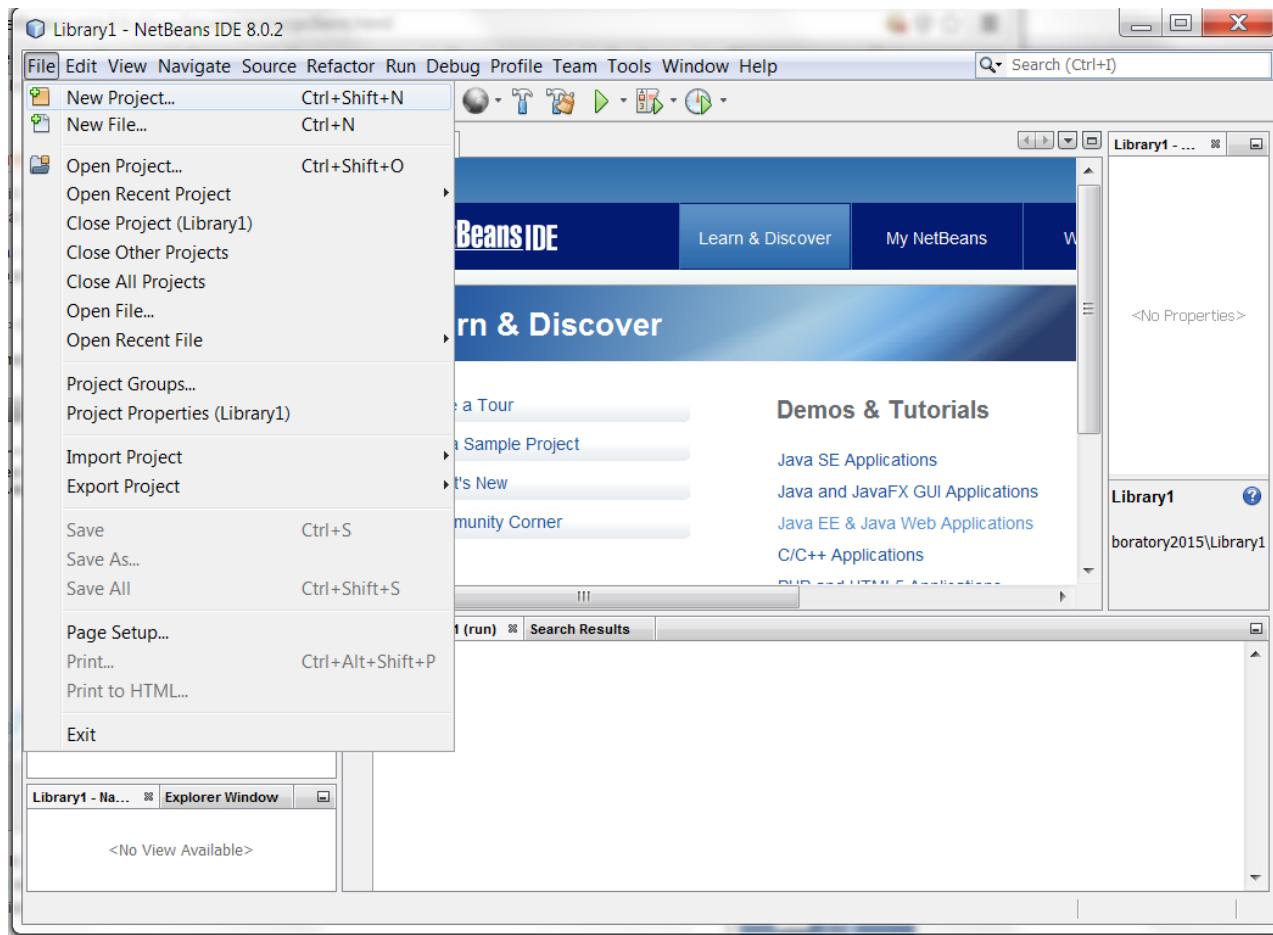
    @Override
    public String Search_accessible_book(String data1[], Object data2) {
        return facade.Search_accessible_book(data1, data2);  }

}
```

5.3.3. The Implementation of FacadeRemote interface as the Facade SessionBean



6. /6.1. /6.1.1. Creation of the EE client tier – click File/New Project...





6.1.2. Creation of the EE client tier – select the Java EE/ Enterprise Application Client items. Click Next.

New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- UML
- Java Web
- Java EE
- HTML5
- Java ME Embedded
- Java Card
- Maven
- PHP
- Groovy

Projects:

- Enterprise Application
- Enterprise Application with Existing Sources
- EJB Module
- EJB Module with Existing Sources
- Enterprise Application Client**
- Enterprise Application Client with Existing S

Description:

Creates a new Enterprise application client project in a standard IDE project. Standard projects use an IDE-generated Ant build script to build, run, and debug the project.

< Back **Next >** Finish Cancel Help





6.1.3. Creation of the EE client tier – fill the project name and select the project location. Click Next.

New Enterprise Application Client X

Steps	Name and Location
1. Choose Project	Project Name: <input type="text" value="Library1_client1-ejb"/>
2. Name and Location	Project Location: <input type="text" value="C:\EnglishLecture\Kruczkiewicz\Laboratory2015"/> <input type="button" value="Browse..."/>
3. Server and Settings	Project Folder: <input type="text" value="Lecture\Kruczkiewicz\Laboratory2015\Library1_client1-ejb"/>
	<input type="checkbox"/> Use Dedicated Folder for Storing Libraries Libraries Folder: <input type="text"/> <input type="button" value="Browse..."/> <small>Different users and projects can share the same compilation libraries (see Help for details).</small>

[< Back](#) [Next >](#) [Finish](#) [Cancel](#) [Help](#)





**6.1.4. Creation of the EE client tier – select the server, Java EE version and fill the name of main class.
Click Finish.**

New Enterprise Application Client

X

Steps	Server and Settings
1. Choose Project 2. Name and Location 3. Server and Settings	Add to Enterprise Application: <None> Server: GlassFish Server 4.1 Add... Java EE Version: Java EE 7 Main Class: client_tier.Client

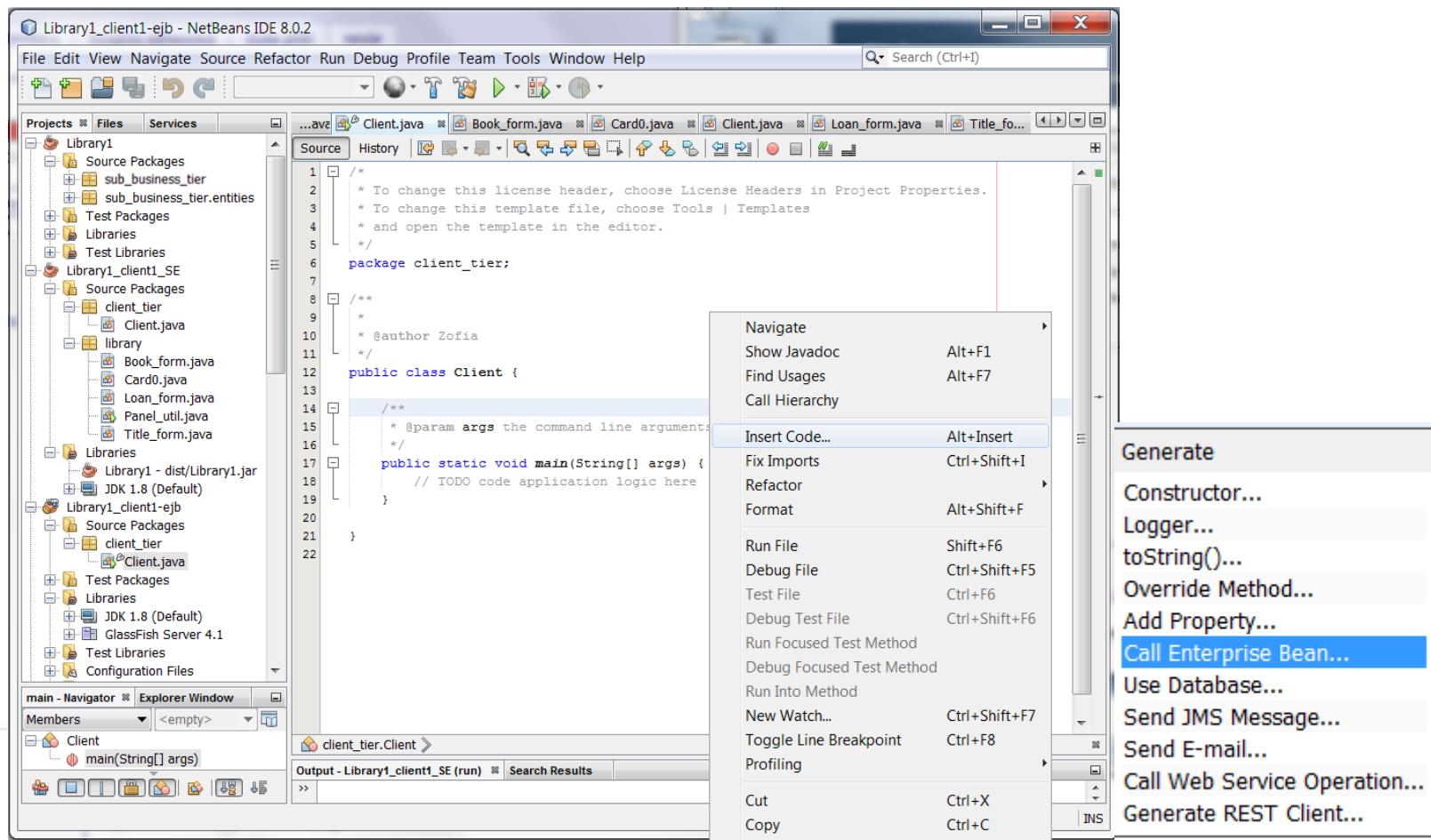
< Back Next > **Finish** Cancel Help





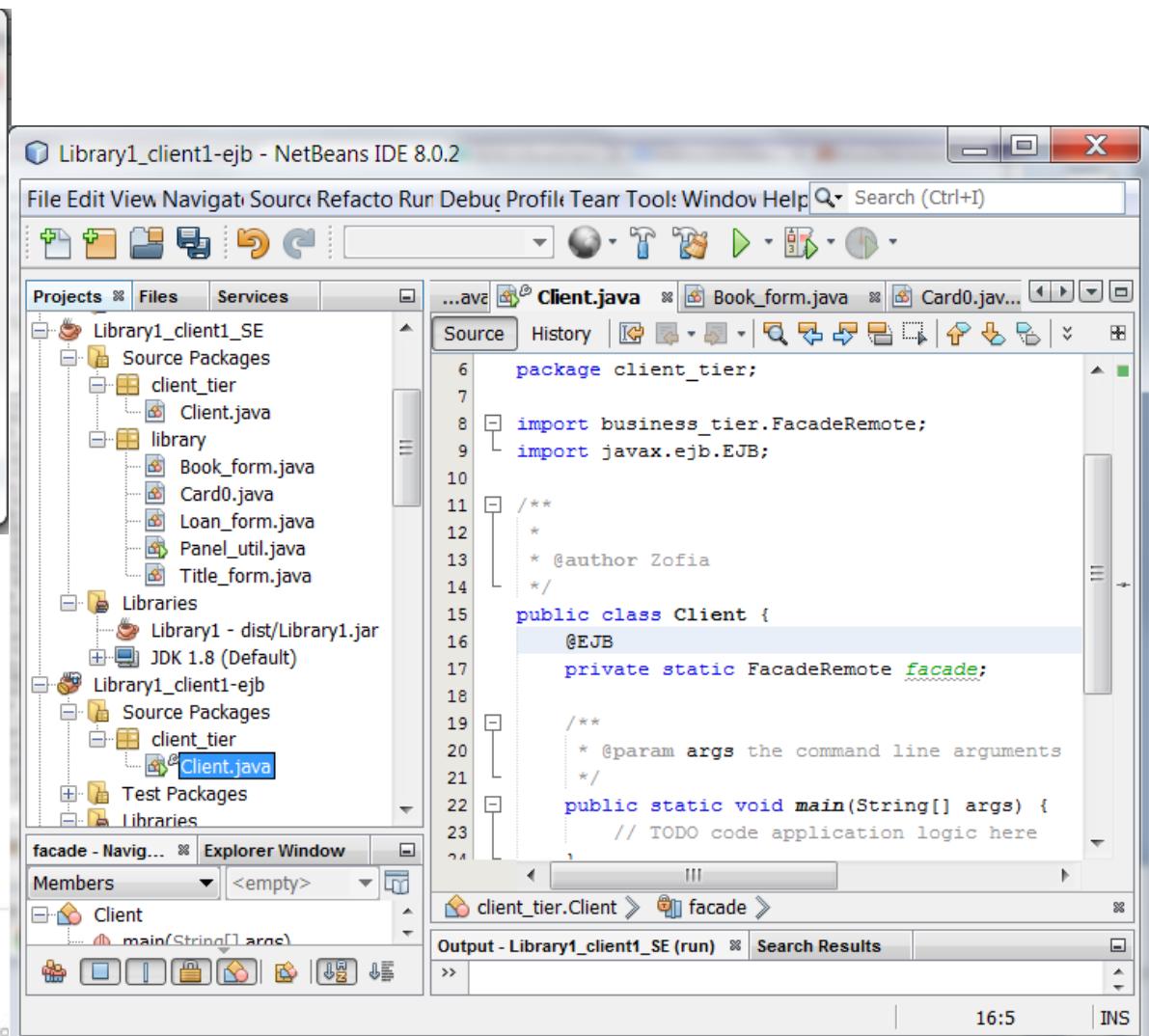
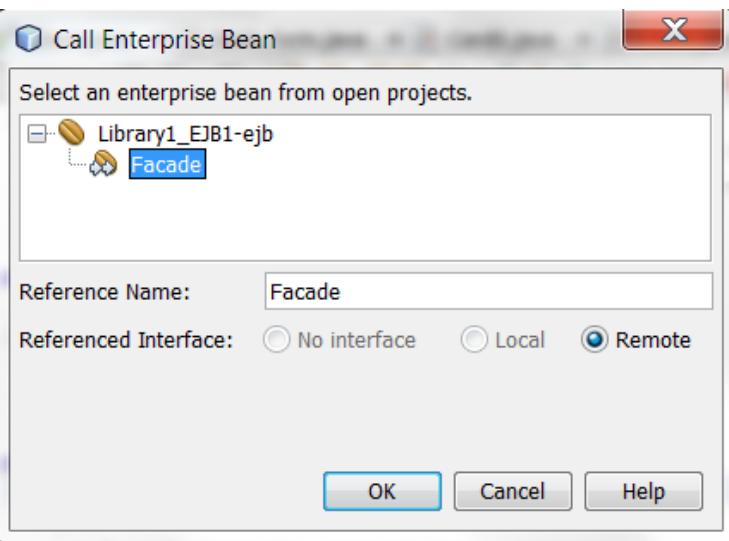
6.2. / 6.2.1. Definition of code of the EE client based on code of the SE client – only change of TFacade class from POJO SE TFacade class into the EJB Facade class.

Creation of connection from the Facade Session Bean of Library1_EJB1-ejb module to the Client class of the Library1_client1-ejb project – expand the Source Packages node of the Library1_client1-ejb project and open Client.java in the editor. Right-click in the source code and choose Insert Code... item and select Call Enterprise Bean to open the Call Enterprise Bean dialog.





6.2.2. Creation of connection from the Facade Session Bean of Library1_EJB1-ejb module to the Client class of the Library_client-ejb project - select the Facade EJB from the Library1_EJB1-ejb module of Library1_EJB1 project



6.2.3. The result of creation the connection to the Facade SessionBean of the Library1_EJB1-ejb module.

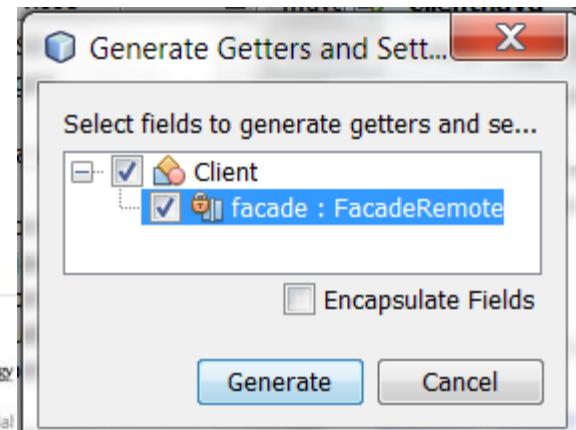
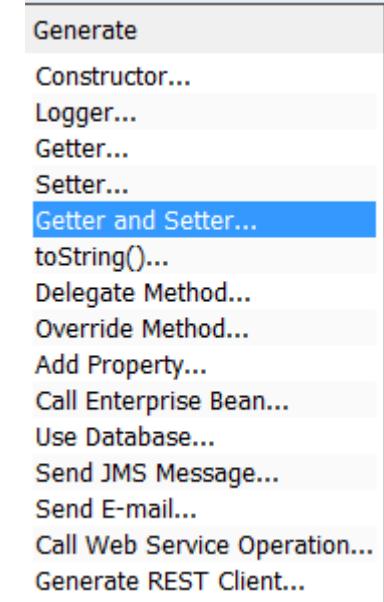




6.3. /6.3.1. Adding the getter and setter methods to the facade attribute as the reference to the Facade SessionBean by using the „Insert Code” process – select Insert Code/Getter and Setter... items to open the Generate Getters and Setters dialog and select the facade attribute

The screenshot shows the NetBeans IDE interface with the title bar "Library1_client1-ejb - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigator, Source, Refactor, Run, Debug, Profiler, Team, Tool, Window, Help, and Search (Ctrl+I). The toolbar has icons for New Project, Open Project, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The Projects tab shows a hierarchy of packages: Library1_client1_SE, client_tier, library, Libraries, and Test Packages. The client_tier package contains Client.java. The library package contains Book_form.java, Card0.java, Loan_form.java, Panel_util.java, and Title_form.java. The Libraries section lists Library1 - dist/Library1.jar and JDK 1.8 (Default). The Test Packages section lists Library1_client1-ejb. The Explorer Window shows the members of Client.java, including a main method and a facade attribute. The Client.java code in the editor is:

```
package client_tier;
import business_tier.FacadeRemote;
import javax.ejb.EJB;
public class Client {
    @EJB
    private FacadeRemote facade;
}
```





6.3.2. The result

The screenshot shows the NetBeans IDE interface with the title bar "Library1_client1-ejb - NetBeans IDE 8.0.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a Search field. The toolbar has icons for file operations like New, Open, Save, and Build. The left pane shows the Project Explorer with two projects: "Library1_client1_SE" and "Library1_client1-ejb". The "client_tier" package contains "Client.java". The "library" package contains "Book_form.java", "Card0.java", "Loan_form.java", "Panel_util.java", and "Title_form.java". The "Libraries" section lists "Library1 - dist/Library1.jar" and "JDK 1.8 (Default)". The "Test Packages" section lists "Library1_interface1-ejb - di...". The "Libraries" section also lists "GlassFish Server 4.1". The right pane displays the "Client.java" code:

```
6 package client_tier;
7
8 import business_tier.FacadeRemote;
9 import javax.ejb.EJB;
10
11 /**
12 * ...
13 * @author Zoltan
14 */
15 public class Client {
16     @EJB
17     private static FacadeRemote facade;
18
19     /**
20      * @param args the command line arguments
21     */
22     public static void main(String[] args) {
23         // TODO code application logic here
24     }
25
26     public static FacadeRemote getFacade() {
27         return facade;
28     }
29
30     public static void setFacade(FacadeRemote facade) {
31         Client.facade = facade;
32     }
33 }
```

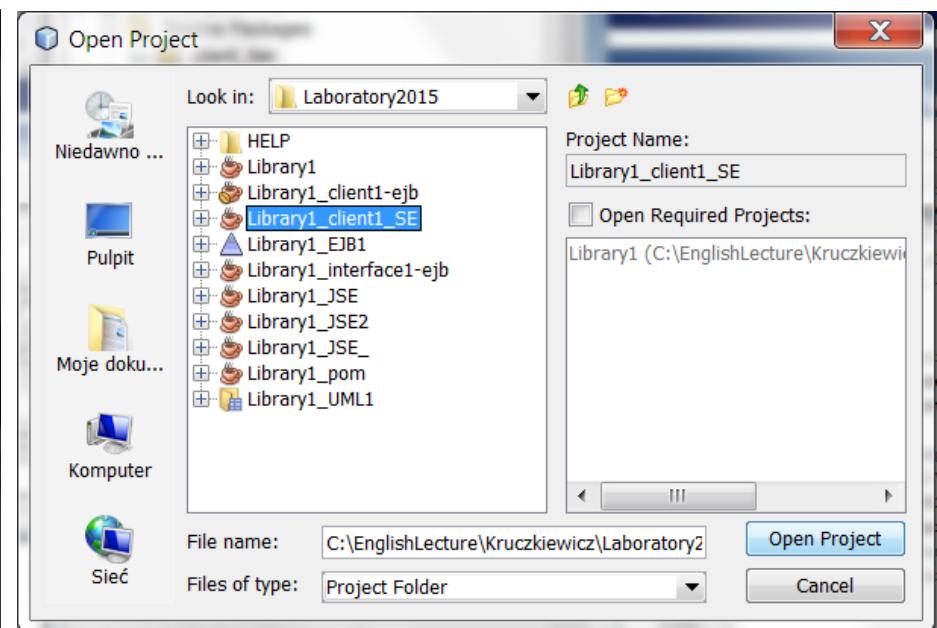
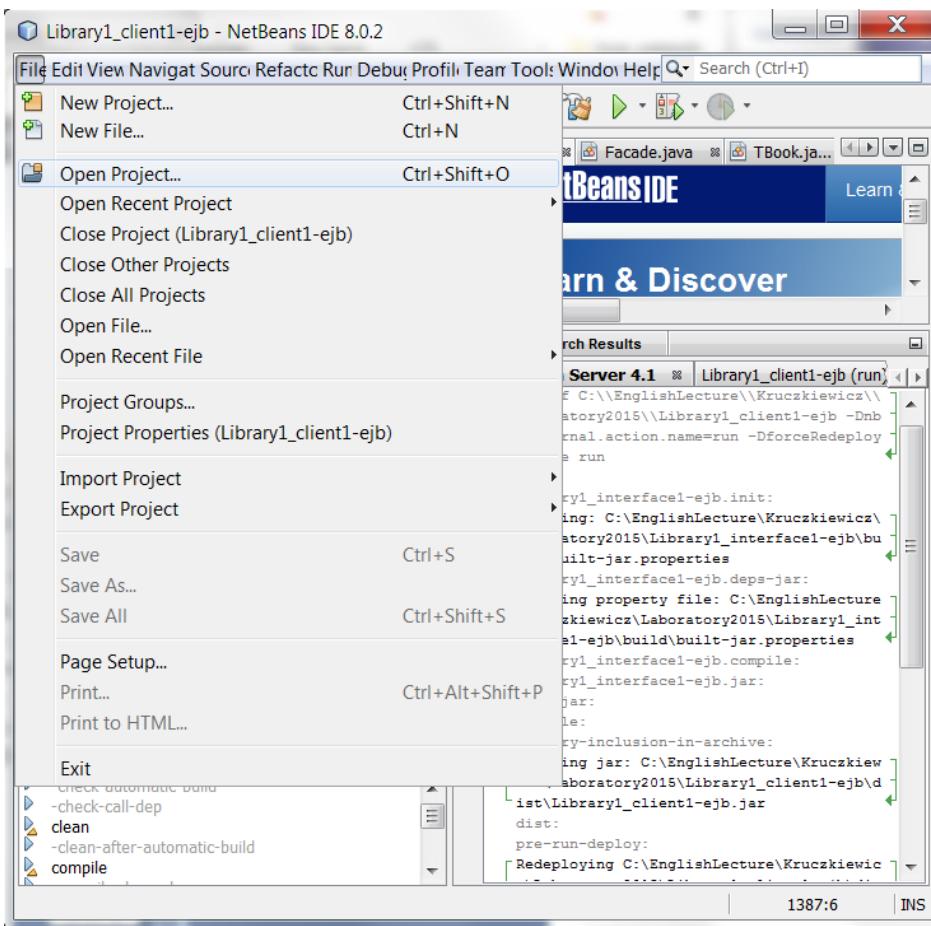
The code defines a class named "Client" with a static EJB reference to "FacadeRemote". It contains a "main" method and two static methods: "getFacade" and "setFacade". The "facade" field is annotated with "@EJB". The "Client" class is located in the "client_tier" package.

6.3.3. Define the body of main method based on code of the Client class main method from the Standard Edition client project named Library1_client1_SE

```
public static void main(String[] args) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            Panel_util.createAndShowGUI();
        }
    });
}
```

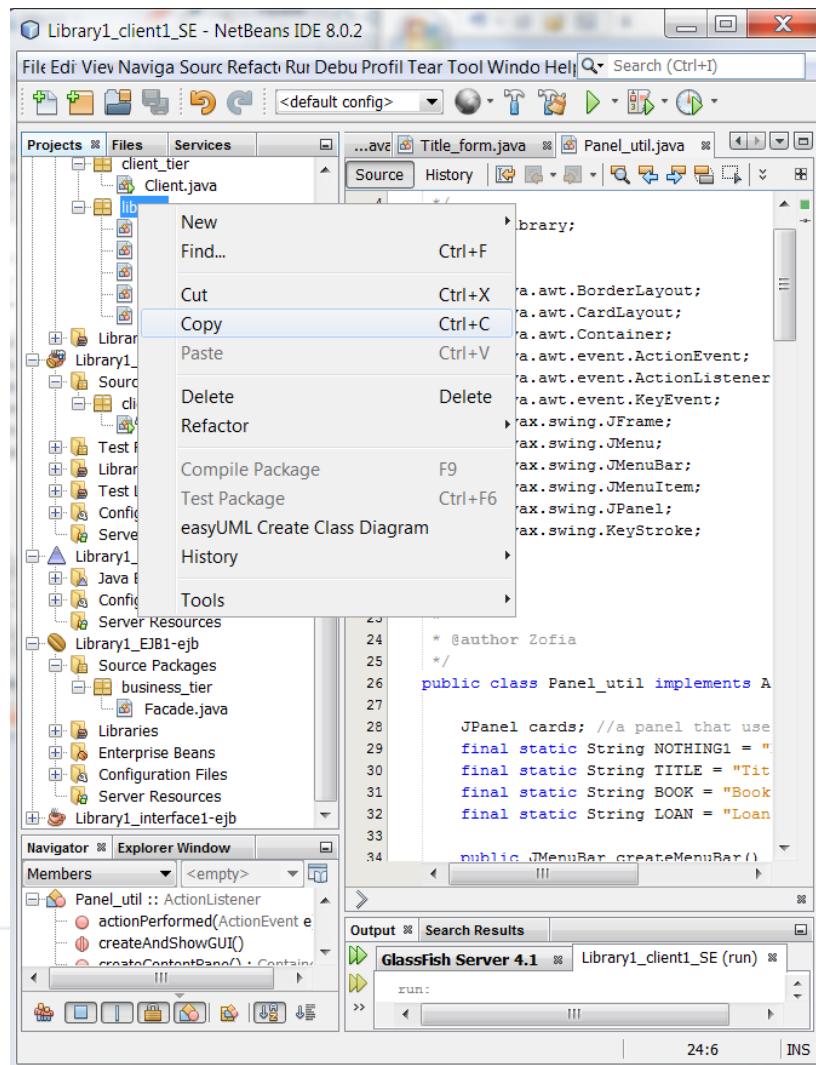


6.4. /6.4.1. Coping the library package from the Standard Edition Application Client project as the Library1_client1_SE one (attachment to the lab3)

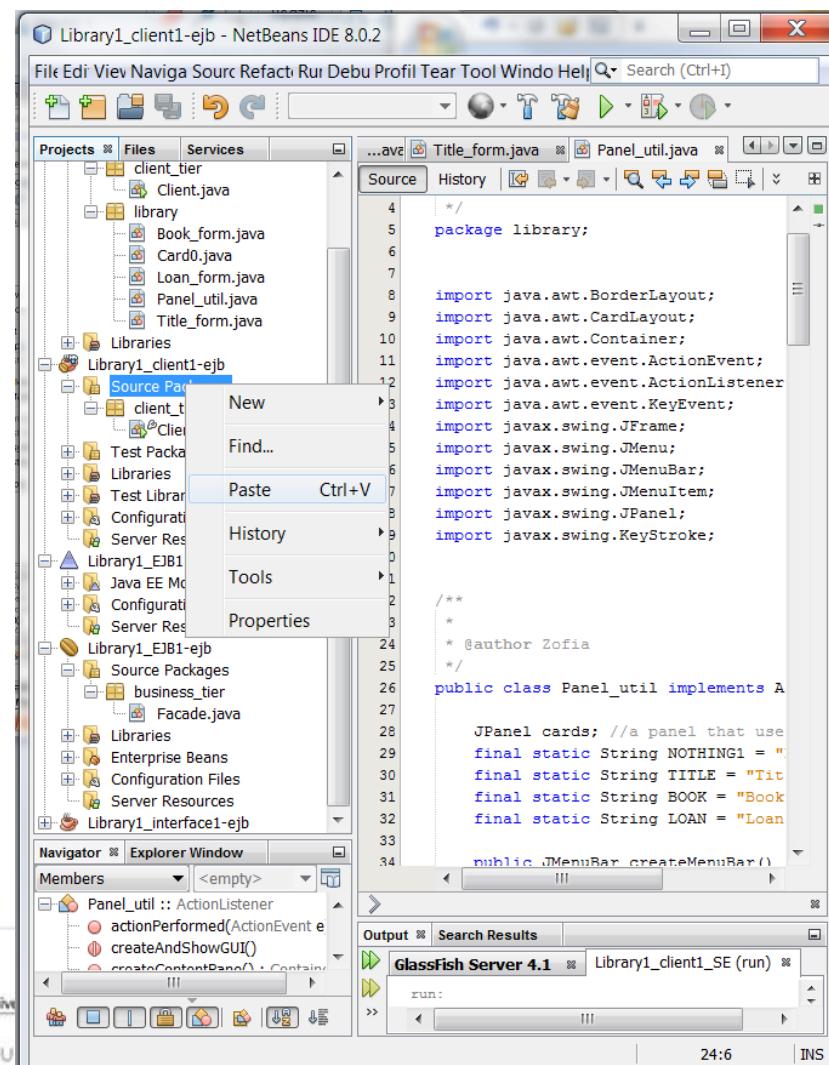




6.4.2. Coping the library package from the Standard Edition Application Client project as the Library1_client1_SE one



6.4.3. Paste the library package from the Library1_client1_SE to the Library1_client1-ejb type Enterprise Application Client project





6.4.4. The result – creation the Library1_client1-ejb type the Enterprise Application Client project based on the code of the Standard Edition version of the client project

Library1_interface1-ejb - NetBeans IDE 8.0.2

File Edit View Naviga Sourc Refact Rui Debu Profil Tear Tool Windo Help Search (Ctrl+I)

Projects Files Services

Source History

```
4  /*
5   * package library;
6   *
7   */
8 import java.awt.BorderLayout;
9 import java.awt.CardLayout;
10 import java.awt.Container;
11 import java.awt.event.ActionEvent;
12 import java.awt.event.ActionListener;
13 import java.awt.event.KeyEvent;
14 import javax.swing.JFrame;
15 import javax.swing.JMenu;
16 import javax.swing.JMenuBar;
17 import javax.swing.JMenuItem;
18 import javax.swing.JPanel;
19 import javax.swing.KeyStroke;
```

Source Packages

client_tier

Client.java

library

Book_form.java

Card0.java

Loan_form.java

Panel_util.java

Title_form.java

Libraries

Library1_client1-ejb

Source Packages

client_tier

Client.java

library

Book_form.java

Card0.java

Loan_form.java

Panel_util.java

Title_form.java

Test Packages

Libraries

Test Libraries

Configuration Files

Server Resources

Library1_EJB1

Java EE Modules

Configuration Files

Server Resources

Navigator Explorer Window

<No View Available>

Output Search Results

GlassFish Server 4.1 Library1_client1_SE (run)

run:

1:1 INS



6.5. The change of getter and setter methods of the Facade class as the Session Bean type attribute

```
package client_tier;

import library.Panel_util;
import sub_business_tier.TFacade;

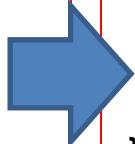
public class Client {

    static TFacade facade = new TFacade();

    static public TFacade getFacade() {
        return facade;
    }

    static public void setFacade(TFacade facade) {
        Client.facade = facade;
    }

    public static void main(String[] args) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Panel_util.createAndShowGUI();
            }
        });
    }
}
```



```
package client_tier;

import business_tier.FacadeRemote;
import javax.ejb.EJB;
import library.Panel_util;

public class Client {
    @EJB
    private static FacadeRemote facade;

    public static FacadeRemote getFacade() {
        return façade;
    }

    public static void setFacade(FacadeRemote facade) {
        Client.facade = façade;
    }

    public static void main(String[] args) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Panel_util.createAndShowGUI();
            }
        });
    }
}
```





6.6. The same code – different type of return value from client.getFacade()

```
public void actionPerformed(ActionEvent evt) {  
    String[] data = form_title();  
    if (data == null) {  
        return;  
    }  
    Client.getFacade().add_title_book(data);  
}
```

```
public void actionPerformed(ActionEvent evt) {  
    String[] data = form_title();  
    if (data == null) {  
        return;  
    }  
    Client.getFacade().add_title_book(data);  
}
```

Object POJO in SE
Client tier

EJB in EE Client
tier

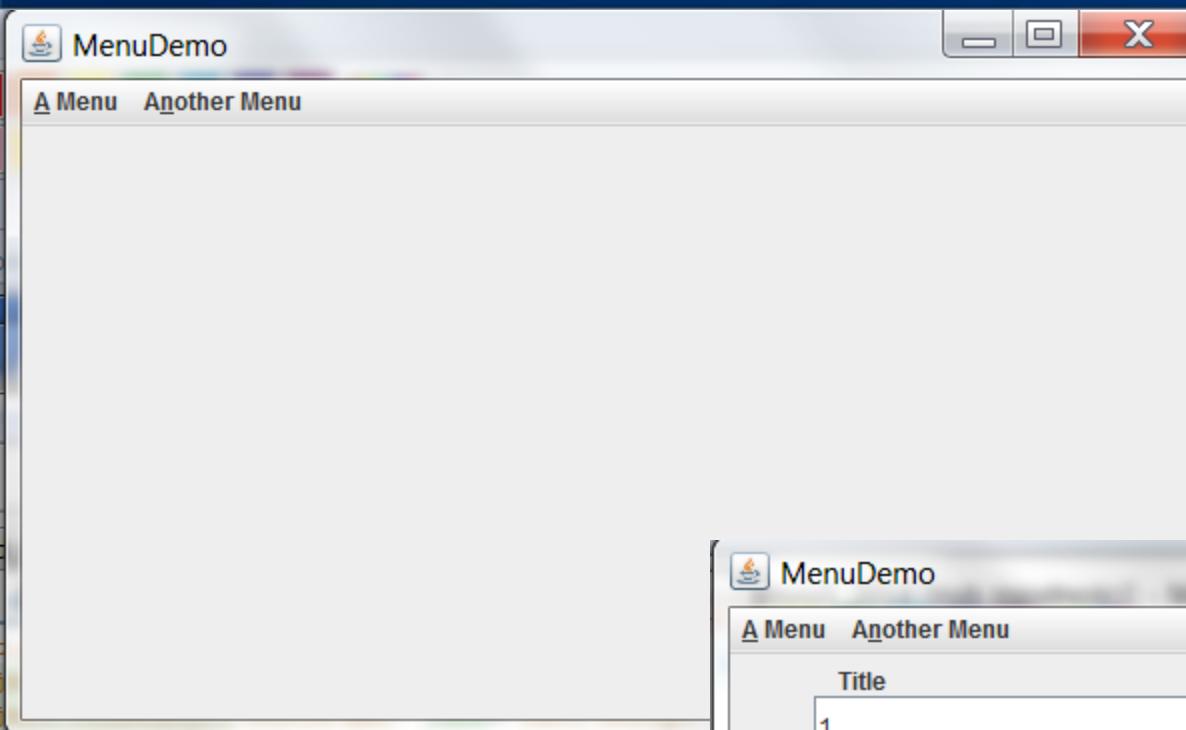




7.0. Running the program

1. After this development, your program will be executed properly, if you lean and build the following programs:
 1. Library1
 2. Library_interface-ejb
 3. Library_EJB1-ejb
 4. Library_client-ejb
2. Then you must deploy the Library_EJB1 program.
3. Finally, you may run a few instances of Library_client-ejb programs - these programs share the common data as titles and books.
4. In the Service Tab you may see, if your EE program deploy properly (Server item). The other useful information you may get from the Glassfish output window tab.





7.1. The view of the Enterprise Application Client (**Library_client-ejb**) for processing of application data - with the same responsibilities as of the version of third laboratory.

Title	1
Author	1
ISBN	1
Publisher	1
Actor	

Add title



7.2. The **Library_client-ejb** form to adding the new books of the selected title, as the application data.

MenuDemo

A Menu Another Menu

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Number of a book

Period of a book

Add book

Books

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 3 Period: Sun Feb 22 22:00:43 CET 2015 ▾

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 3 Period: Sun Feb 22 22:00:43 CET 2015

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 5





7.3. Restored data from database, after again opening EE application with the one kind of client:
Library_client-ejb as the Enterprise Application client (below)

MenuDemo

A Menu Another Menu

Publisher	ISBN	Title	Author	Actor
1	1	1	1	
1	1	1	1	1

Number of a book

Period of a book

Add book

Books

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 3 Period: Sun Feb 22 22:00:43 CET 2015

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 3 Period: Sun Feb 22 22:00:43 CET 2015

Title: 1 Author: 1 ISBN: 1 Publisher: 1 Actor: 1 Number: 5

