

Laborator 9

1. Instalarea si configurarea mediului de dezvoltare (20 min)

Pentru a se putea lucra cu EJB s-a ales solutia bazata pe Eclipse si server GlassFish. Exista si alte abordari cum ar fi IBM VisualAge® for Java sau Oracle JDeveloper sau abordarile bazate pe Eclipse si Apache.

Pentru instalarea de la zero ("clean install") se vor efectua urmatoarele operatii

- Se in staleză Eclipse Kepler 2 de la adresa <https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplersr2>
- Daca nu este deja instalat (se verifica) se va instala JDK 7 de la <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>.
- Se va descarca GlassFish Server Open Source Edition 4.0 de la
 - <https://glassfish.java.net/download.html> sau
 - <http://download.java.net/glassfish/4.0/release/glassfish-4.0.zip>.Si se va despachetea fișierul zip (de exemplu în c:\).
- Se va instala Oracle Enterprise Pack for Eclipse cu urmatoarea suită de operatii
 - În Eclipse: **Help** → **Install New Software...** → **Add...** → **Location** → **url:** http://download.oracle.com/otn_software/oepe/kepler →
 - Apoi de la Oracle Enterprise Pack selectează **GlassFish Tools** → **Next** → ...

2. Creerea aplicației EJB

Se va incepe prin crearea unui proiect EJB folosind urmatoarea suită de operatii in Eclipse

- **File** → **New** → **EJB Project** → **nume proiect** (de exemplu HelloWorld) → **New runtime...** → selectează **GlassFish 4.0** → **Next** → selectează **JDK 7** din primul drop down * → **Browse...**
- pentru a selecta **Glassfish Server Directory** → navighează la directorul glassfish unde s-a dezarhivat glassfish4 (c:\glassfish4\glassfish) → **Finish** → **Next** → **Next** → bifează opțiunea **Generate ejb-jar.xml deployment descriptor** → **Finish**.
- * - dacă nu este nici o opțiune cu JDK 7 în drop down: click pe **Installed JREs Preferences** → **Add...** → **Next** → **Directory...** → navighează unde s-a instalat jdk-ul (c:\Program Files\Java\jdk1.7.0_51) → **Ok** → **Finish** → bifează opțiunea **jdk1.7.0_51** → **Ok**

3. Pentru a adauga un Enterprise Java Bean la proiect vom efectua urmatoarele operatii:

- click dreapta pe proiectul creat → **New** → **Session Bean (EJB 3.x)** → **Java package: test** → **Class name: HelloBean** → bifează opțiunea **Remote** → **Next** → **Finish**

4. O data creata sursa bean-ul se vor face urmatoarele modificari

- Se va sterge instructiunea **@LocalBean** din clasa **HelloBean**
- În **HelloBeanRemote.java** adaugă declarația: **public String sayHello();**
- În **HelloBean.java** adaugă definiția: **public String sayHello() { return "Hello World!"; }**
- Click dreapta pe proiectul **HelloWorld** → **Run As** → **Run on Server** → **Next** → **Next** → **Finish**
- Click dreapta pe proiectul **HelloWorld** → **Export** → **EJB JAR File** → alege o destinație (c:\pp\HelloWorld.jar) → **Finish**

5. Creează aplicația client

- **File** → **New** → **Other...** → **Java Project** → **Next** → **Project name: HelloWorldClient** → **Next** → în tab-ul **Libraries** → **Add External JARs** → selectează fișierul **HelloWorld.jar** creat anterior → **Open** → **Add External JARs** → navighează în directorul glassfish4 → **glassfish\lib\appserv-rt.jar** (C:\glassfish4\glassfish\lib\appserv-rt.jar) → **Open** → **Finish** → (dacă apare o fereastră dă click pe **No**)
- Click dreapta pe proiectul **HelloWorldClient** → **New** → **Class** → **Name: HelloTest** → **Finish**

Tema 1. Testați urmatoarea aplicație EJB (30 min).

```
import java.util.Properties;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import test.HelloBeanRemote;
public class HelloBeanClient
{
```

```

public static void main(String[] args) throws NamingException
{
    Properties jndiProps = new Properties();
    jndiProps.put("java.naming.factory.initial", "com.sun.enterprise.naming.impl.Serial
InitContextFactory");
    jndiProps.put("java.naming.factory.url.pkgs", "com.sun.enterprise.naming");
    jndiProps.put("java.naming.factory.state", "com.sun.corba.ee.impl.presentation.rmi.
JNDIStateFactoryImpl");
    jndiProps.setProperty("org.omg.CORBA.ORBInitialHost", "127.0.0.1");
    jndiProps.setProperty("org.omg.CORBA.ORBInitialPort", "3700");
    Context ctx = new InitialContext(jndiProps);
    Object obj = ctx.lookup("java:global/HelloWorld/HelloBean");
    HelloBeanRemote bean = (HelloBeanRemote) obj;
    System.out.println(bean.sayHello());
}
}

```

Tema 2. Creați o aplicație EJB care să permită gestionarea cărților dintr-o bibliotecă pornind de la următoarele interfețe (40 min):

```
import java.io.Serializable;
```

```
import javax.ejb.Remote;
```

```
@Remote
```

```
public interface CarteBeanRemote extends Serializable
```

```

{
    public String getTitlu();
    public String getAutor();
    public int getAnAparitie();
    public String getIsbn();
    public String detalii();
}

```

```
import java.util.List;
```

```
import javax.ejb.Remote;
```

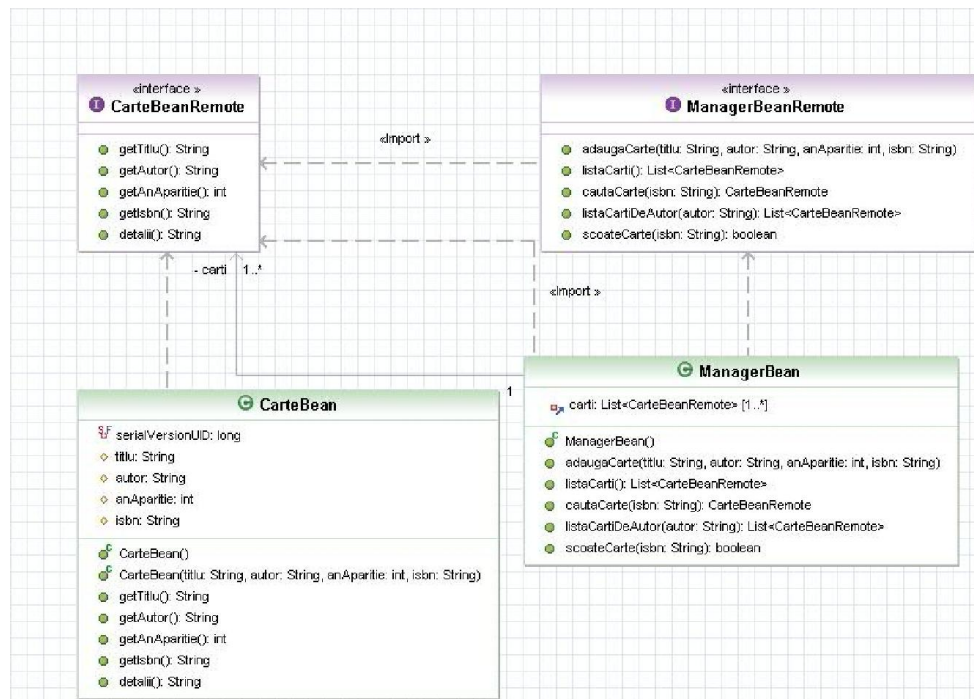
```
@Remote
```

```
public interface ManagerBeanRemote
```

```

{
    public void adaugaCarte(String titlu, String autor, int anAparitie, String isbn);
    public List<CarteBeanRemote> listaCarti();
}

```



Tema 3. Creați o aplicație client pentru a testa aplicația EJB de mai sus (10 min).

Tema pe acasa. Adăugați metode care să permită:

- căutarea unei cărți după isbn,
- afișarea tuturor cărților scrise de un anumit autor sau publicate într-un anumit an,
- împrumutarea unor cărți precum și returnarea lor,
- scoaterea unor cărți din bibliotecă,
- creați un meniu care să permită selectarea operațiilor enumerate mai sus.