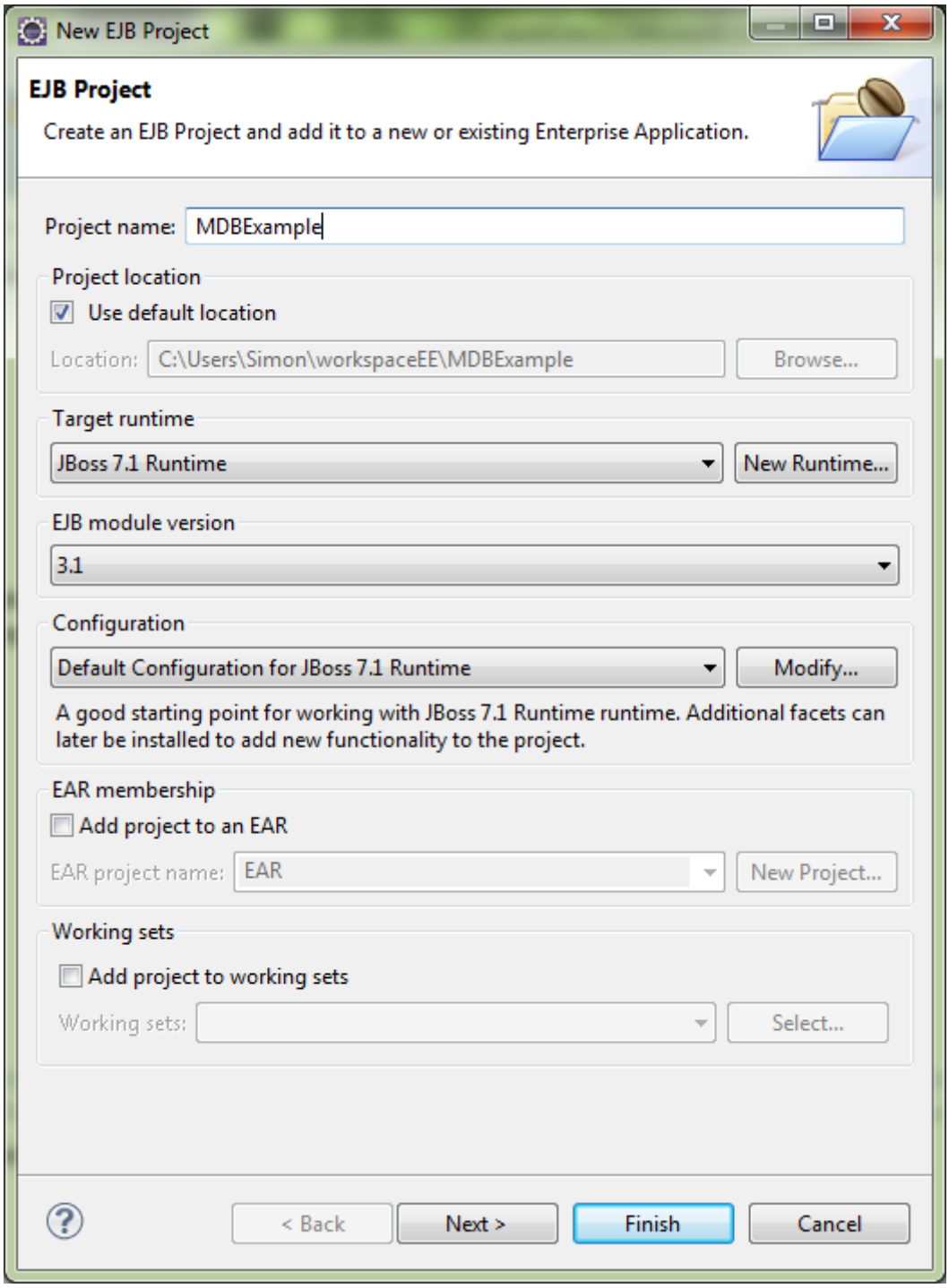
Message Driven Bean

Készítsünk egy üzenet feldolgozó beant.

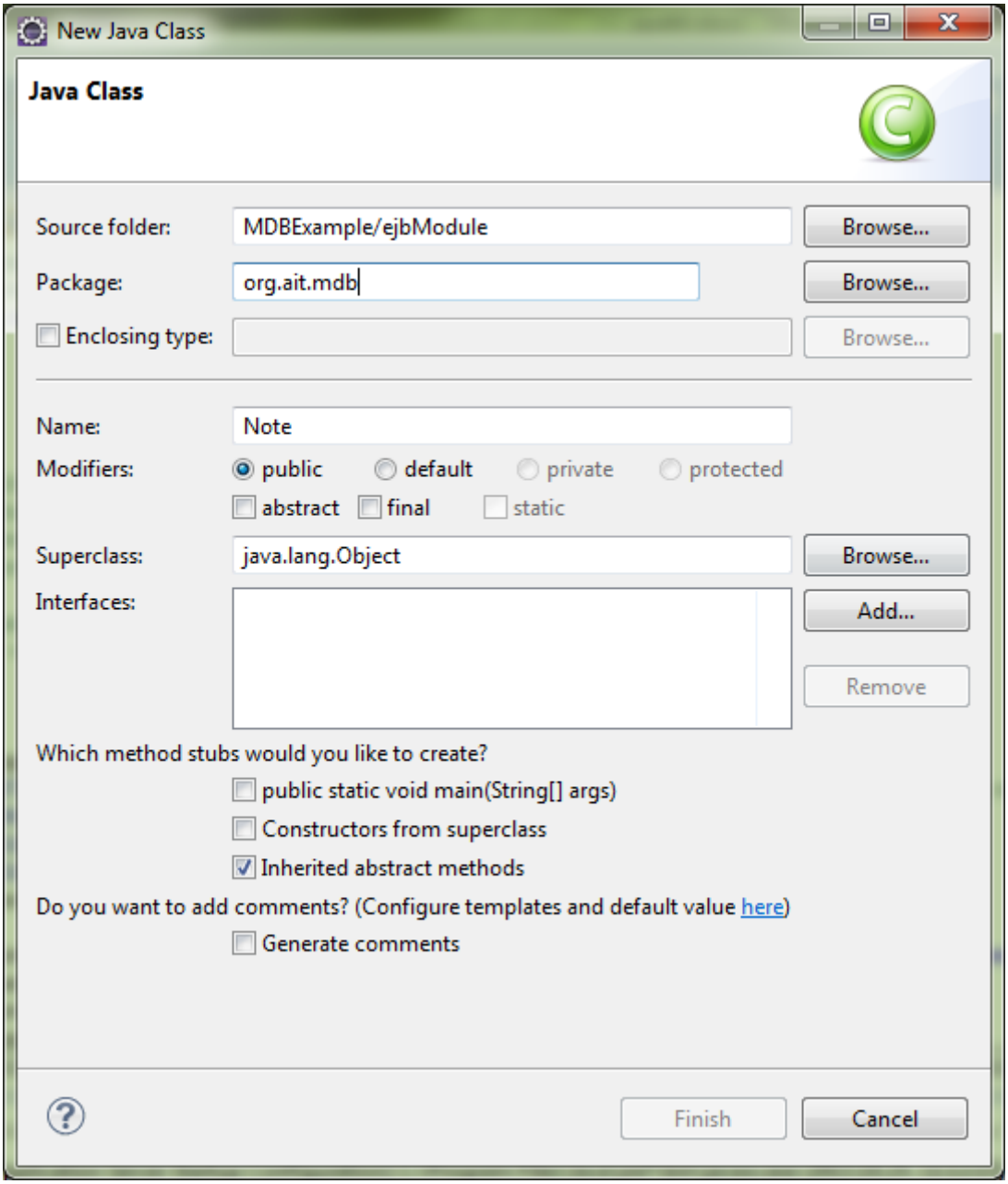
Hozzunk létre egy EJB projectet

File → New → EJB Project
A megjelenő párbeszédablakban adjuk meg a projekt nevét, ez a példában MDBExample, majd kattintsunk a *Finish* gombra.



Üzenet osztály létrehozása

Hozzunk létre egy üzenet osztályt, amit továbbíthatunk majd az üzenet vezérelt bennek. A neve legyen *Note*.



A *Note* osztály törzse legyen a következő:

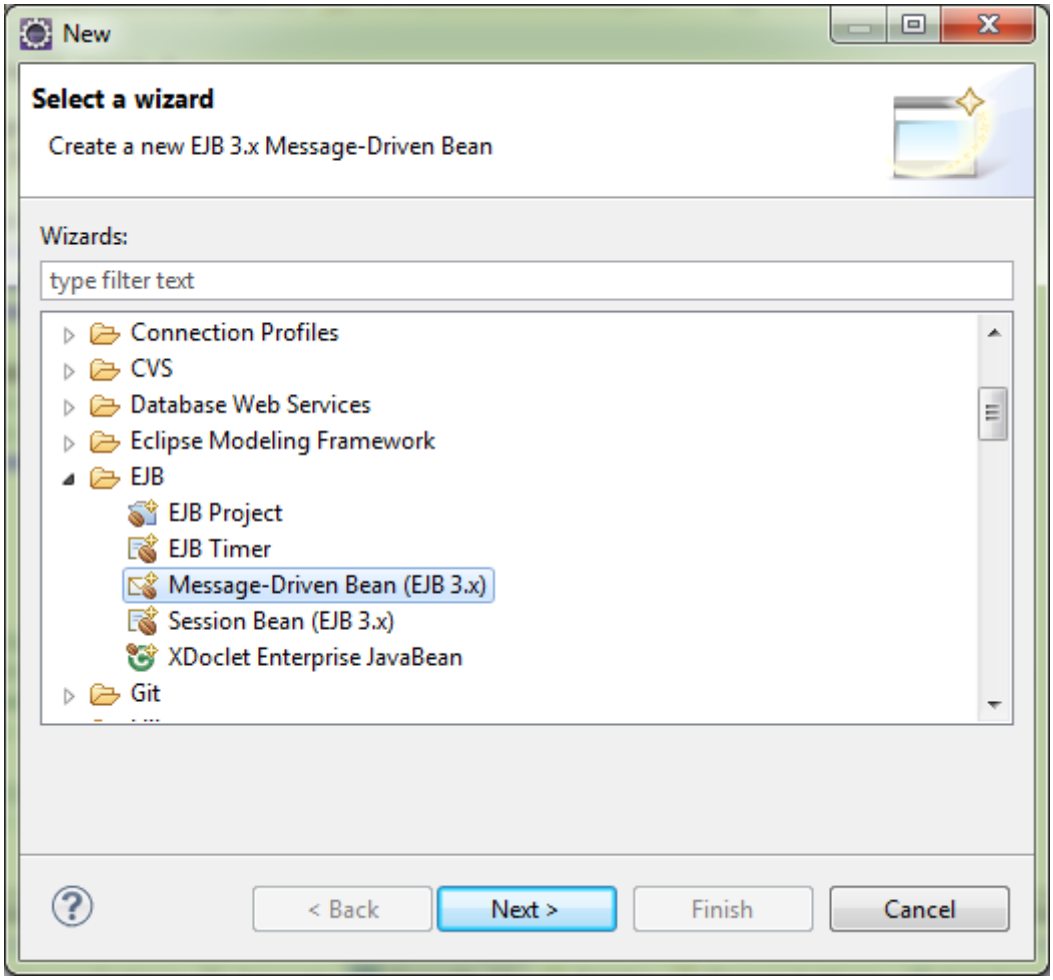
```
package org.ait.mdb;

import java.io.Serializable;

public class Note implements Serializable {
	private int id;
	private String text;
	public Note() {
```

```
	}

	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getText() {
		return text;
	}

	public void setText(String text) {
		this.text = text;
	}

}
```

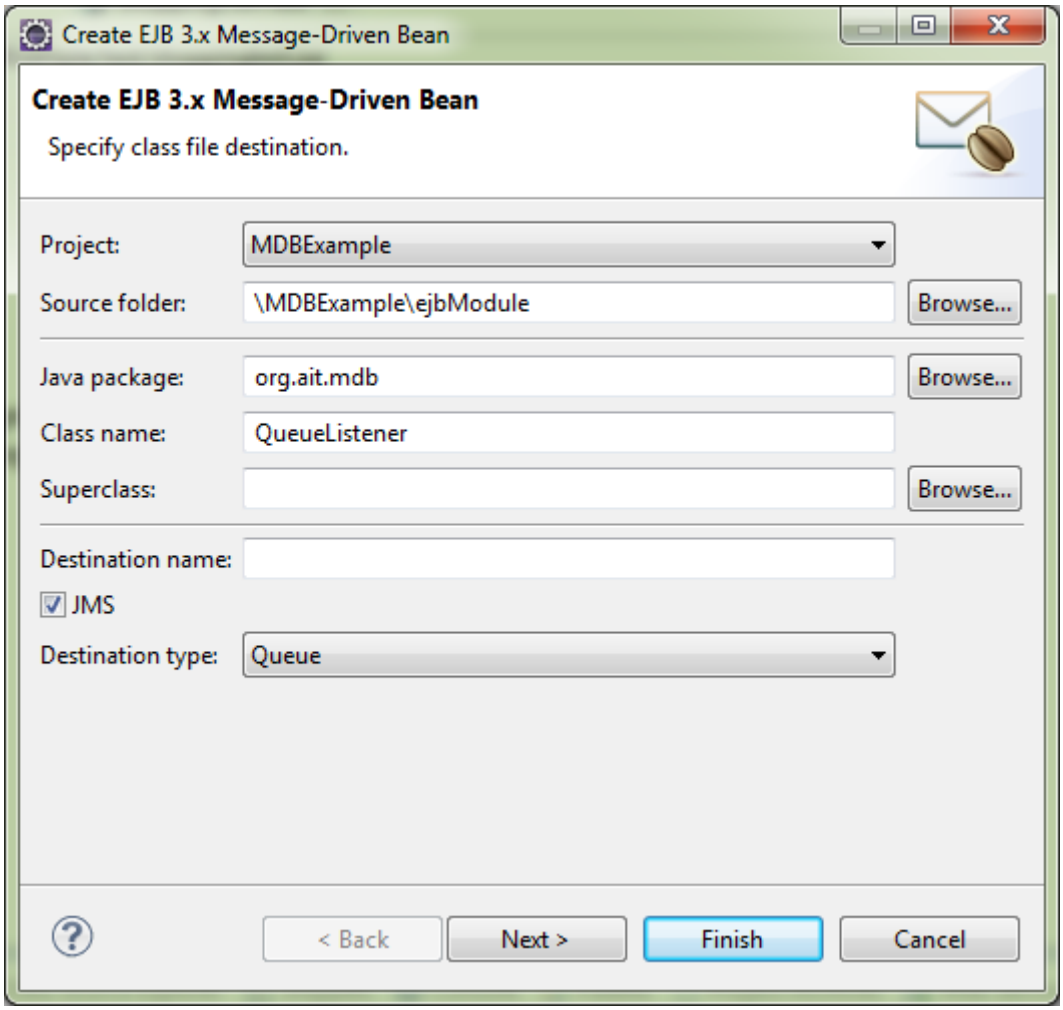
Ezután hozzuk létre a message-driven beant, ami majd az üzeneteket feldolgozza.

Message-Driven Bean létrehozás

Új Message-Driven Bean létrehozását a következő menüpontban lehet megtenni:
File → New → Other … itt a következőket kell választani: EJB → Message-Driven Bean



A megjelenő párbeszédablakban a bean paraméterei állíthatók be.
A példában a csomag név *org.ait.mdb*, az osztály neve pedig *QueueListener*.
A *Destination type*-ot állítsuk *Queue*-re.
Majd kattintsunk a *Next* gombra, majd a *Finish*-re.



A *QueueListener* törzse legyen a következő:

```
import javax.jms.ObjectMessage;
import javax.jms.TextMessage;


/**
 * Message-Driven Bean implementation class for: QueueListener
 */
@MessageDriven(
		activationConfig = {
			@ActivationConfigProperty(
					propertyName = "destinationType", propertyValue =
"javax.jms.Queue"),
			@ActivationConfigProperty(
					propertyName = "destination", propertyValue =
"queue/MyQueue")
		})
public class QueueListener implements MessageListener {

    /**
     * Default constructor.
     */
    public QueueListener() {

    }

      /**
     * @see MessageListener#onMessage(Message)
     */
    public void onMessage(Message message) {
        try {
			if(message instanceof TextMessage) {
				System.out.println("Queue: I received a TextMessage at " + new
Date());
				TextMessage msg = (TextMessage) message;
				System.out.println("Az üzenet: " + msg.getText());
			} else if(message instanceof ObjectMessage) {
				System.out.println("Queue: I received a TextMessage at " + new
Date());
				ObjectMessage msg = (ObjectMessage) message;
				Note note = (Note) msg.getObject();
				System.out.println("Note details:");
				System.out.println(note.getId());
				System.out.println(note.getText());
			} else {
				System.out.println("Not valid message");
			}
        } catch(JMSException e) {
			e.printStackTrace();
        }

    }

}
```

Bean hozzáadása szerverhez

Jobb gombbal kattintsunk a projekten majd → *Run As* → *Run On Server* pontotra kattintva válasszuk a „*JBoss 7.1 Runtime Server*”-t és kattintsunk a *Finish* gobra.

JBoss AS 7 üzenetküldő szolgáltatásának beállítása

A beállításokhoz a szerver beállítás fájlját kell szerkeszteni, ami a *standalone.xml* és a *JbossAS_Home/standalone/configuration* könyvtárban található.

Az <*extension*> elemhez hozzá kell adni a következőt:

```
<extension module="org.jboss.as.messaging"/>
```

A következő alrendszert hozzá kell adni a <*profile*> -ban lévő <*subsystem*> elemhez:

```
<subsystem xmlns="urn:jboss:domain:messaging:1.1">
            <hornetq-server>
                <persistence-enabled>true</persistence-enabled>
                <journal-file-size>102400</journal-file-size>
                <journal-min-files>2</journal-min-files>

                <connectors>
                    <netty-connector name="netty" socket-binding="messaging"/>
                    <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
                        <param key="batch-delay" value="50"/>
                    </netty-connector>
                    <in-vm-connector name="in-vm" server-id="0"/>
                </connectors>

                <acceptors>
                    <netty-acceptor name="netty" socket-binding="messaging"/>
                    <netty-acceptor name="netty-throughput" socket-
binding="messaging-throughput">
                        <param key="batch-delay" value="50"/>
                        <param key="direct-deliver" value="false"/>
                    </netty-acceptor>
                    <in-vm-acceptor name="in-vm" server-id="0"/>
                </acceptors>

                <security-settings>
                    <security-setting match="#">
                        <permission type="send" roles="guest"/>
                        <permission type="consume" roles="guest"/>
                        <permission type="createNonDurableQueue" roles="guest"/>
                        <permission type="deleteNonDurableQueue" roles="guest"/>
                    </security-setting>
                </security-settings>

                <address-settings>
                    <address-setting match="#">
                        <dead-letter-address>jms.queue.DLQ</dead-letter-address>
                        <expiry-address>jms.queue.ExpiryQueue</expiry-address>
```

```
                        <redelivery-delay>0</redelivery-delay>
                        <max-size-bytes>10485760</max-size-bytes>
                        <address-full-policy>BLOCK</address-full-policy>
                        <message-counter-history-day-limit>10</message-counter-
history-day-limit>
                    </address-setting>
                </address-settings>

                <jms-connection-factories>
                    <connection-factory name="InVmConnectionFactory">
                        <connectors>
                            <connector-ref connector-name="in-vm"/>
                        </connectors>
                        <entries>
                            <entry name="java:/ConnectionFactory"/>
                        </entries>
                    </connection-factory>
                    <connection-factory name="RemoteConnectionFactory">
                        <connectors>
                            <connector-ref connector-name="netty"/>
                        </connectors>
                        <entries>
                            <entry name="RemoteConnectionFactory"/>
                            <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
                        </entries>
                    </connection-factory>
                    <pooled-connection-factory name="hornetq-ra">
                        <transaction mode="xa"/>
                        <connectors>
                            <connector-ref connector-name="in-vm"/>
                        </connectors>
                        <entries>
                            <entry name="java:/JmsXA"/>
                        </entries>
                    </pooled-connection-factory>
                </jms-connection-factories>

                <jms-destinations>
                    <jms-queue name="testQueue">
                        <entry name="queue/MyQueue"/>
                    </jms-queue>
                    <jms-topic name="testTopic">
                        <entry name="topic/MyTopic"/>
                    </jms-topic>
                </jms-destinations>
            </hornetq-server>
        </subsystem>
```

Az üzenetváltó port hozzáadásához a <*socket-binding-group*> -hoz hozzá kell adni a következőt:

```
<socket-binding name="messaging" port="5445"/>
<socket-binding name="messaging-throughput" port="5455"/>
```

A <*subsystem xmlns=”urn:jboss:domain:ejb3:1.2"*> elemhez hozzá kell adni a következő <*mdb*> elemet:

```
<mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

A

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.0"/>
```

elemben lévő

```
<login-module code="RealmUsersRoles" flag="required">
```

elemhez hozzá kell adni a következőt:

```
<module-option name="unauthenticatedIdentity" value="guest"/>
```

```
<subsystem xmlns="urn:jboss:domain:security:1.1">
    <security-domains>
        <security-domain name="other" cache-type="default">
            <authentication>
                <login-module code="Remoting" flag="optional">
                    <module-option name="password-stacking" value="useFirstPass"/>
                </login-module>
                <login-module code="RealmUsersRoles" flag="required">
                    <module-option name="usersProperties"
value="${jboss.server.config.dir}/application-users.properties"/>
                    <module-option name="rolesProperties"
value="${jboss.server.config.dir}/application-roles.properties"/>
                    <module-option name="realm" value="ApplicationRealm"/>
                    <module-option name="password-stacking" value="useFirstPass"/>
                    <module-option name="unauthenticatedIdentity" value="guest"/>
                </login-module>
```

A guest felhasználó létrehozása

A *JbossAS_Home/standalone/configuration* könyvtárban található *application-roles.properties* fájlban el kell távolítani a kommentet jelző # karaktert a *guest=guest* sor elejéről.
Ezzel lehetővé válik azonosítatlan felhasználók számára az üzenetsor elérése.
Abban az esetben ha azonosított felhasználóként szeretnénk elérni az üzenetsort, a *bin* könyvtárban lévő *add-user.bat* fájlt kell elindítani és megadni a felhasználó adatokat.

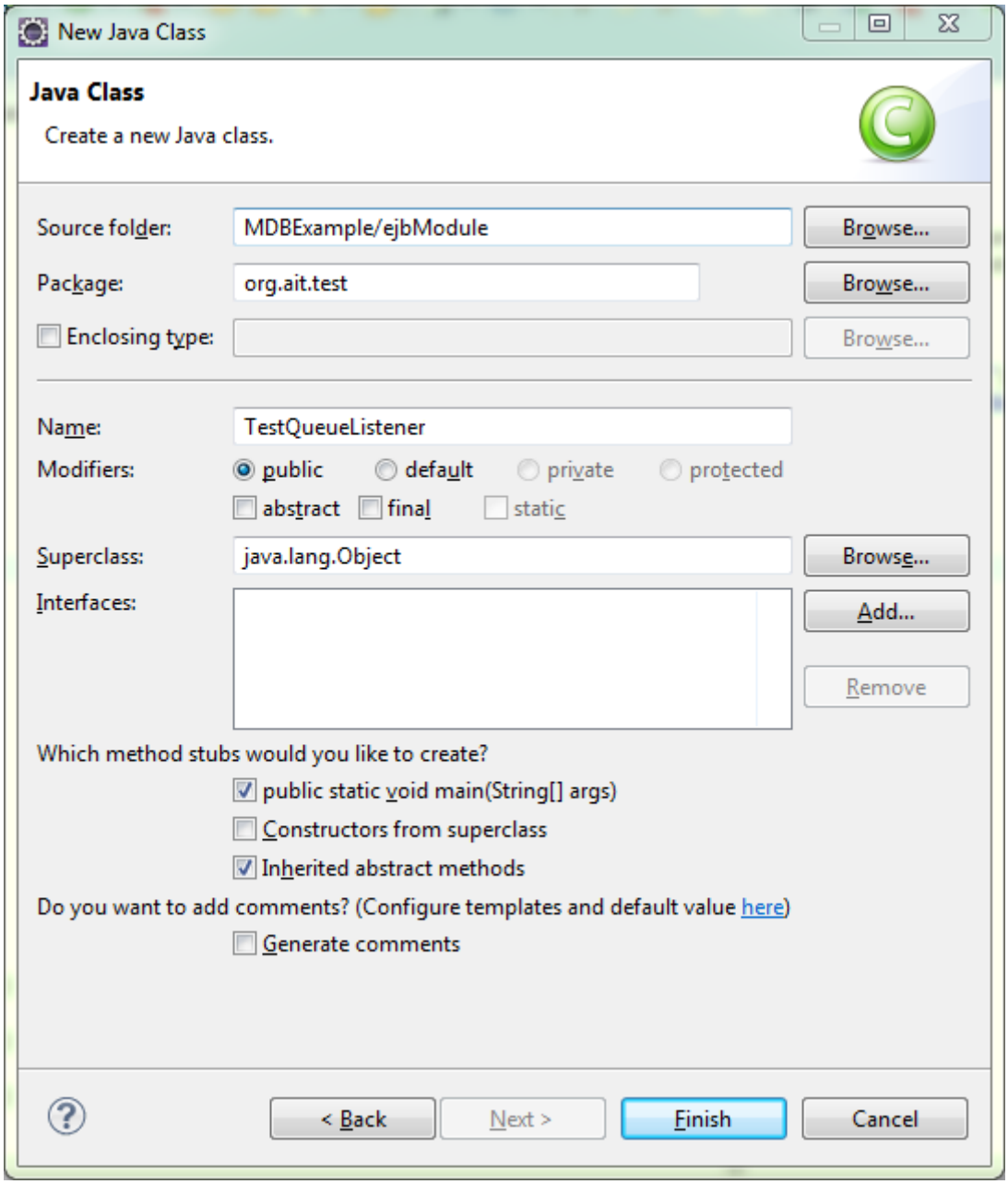
Projekt hozzáadása szerverhez

Miután elkészítettük a beanünket telepíteni kell a szerverre.
Ezt kétféleképpen is megtehetjük:

- Jobb gombbal kattintsunk a projekten majd → *Run As* → *Run On Server* pontotra kattintva válasszuk a „*JBoss 7.1 Runtime Server*”-t és kattintsunk a *Finish* gobra.
- A *Server* fülön klikkeljünk jobb gombbal a *JBoss 7.1 Runtime Server* –re a szerver nézetben, majd → *Add and Remove* → majd válasszuk ki a jobb oldali panelen a projektünket és kattintsunk a *Finish* gombra.

Teszt kliens létrehozása

Hozzunk létre egy osztályt amellyel tesztelhetjük a létrehozott message beant.
Válasszuk a *New* menüben a *Class* menüpontot, majd a megjelenő ablakban adjuk meg az osztály nevét, csomagnevét és jelöljük be a main metódus vázának a legenerálását.



A *TestQueueListener* osztály tartalma legyen a következő:

```
package org.ait.test;

import java.util.Properties;

import javax.annotation.Resource;
import javax.jms.*;
import javax.naming.*

import org.ait.mdb.Note;
import org.ait.mdb.QueueListener;
import org.hornetq.api.core.*;
import org.hornetq.api.core.client.*;
import org.hornetq.api.jms.HornetQJMSClient;
import org.hornetq.api.jms.JMSFactoryType;
import org.hornetq.core.remoting.impl.invm.InVMConnectorFactory;
import org.hornetq.core.remoting.impl.netty.NettyConnectorFactory;

public class TestQueueListener {

	public static void main(String[] args) {

		final String QUEUE_LOOKUP = "queue/MyQueue";
		final String CONNECTION_FACTORY = "ConnectionFactory";

		try {
			TransportConfiguration transportConfiguration = new
TransportConfiguration(NettyConnectorFactory.class.getName());
			ConnectionFactory connectionFactory = (ConnectionFactory)
HornetQJMSClient.createConnectionFactoryWithoutHA(JMSFactoryType.CF,transportConfi
guration);
			//The queue name should match the jms-queue name in
standalone.xml
		Queue queue = HornetQJMSClient.createQueue("testQueue");

			Connection connection = connectionFactory.createConnection();
	    Session session = connection.createSession(false,
QueueSession.AUTO_ACKNOWLEDGE);

	    MessageProducer messageProducer = session.createProducer(queue);

			TextMessage message = session.createTextMessage();
			message.setText("Hello EJB3 MDB Queue!!!");
			messageProducer.send(message);
			System.out.println("Sent TextMessage to the Queue");

			ObjectMessage objMsg = session.createObjectMessage();

			Note note = new Note();
			note.setId(2163);
			note.setText("One note to the Queue.");
			objMsg.setObject(note);
			messageProducer.send(objMsg);
			System.out.println("Sent ObjectMessage to the Queue");

			session.close();

		} catch (JMSException e) {
```

```
                    e.printStackTrace();
              } catch (Exception e) {
                    e.printStackTrace();
              }
        }
}
```

Miután ezzel megvagyunk, futtassuk a teszt osztályunkat alkalmazásként.