Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение наукии высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

**Институт информационных технологий и радиотехники**

**(ИИТР)**

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 05**

**по дисциплине**

**«Распределенные программные системы»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

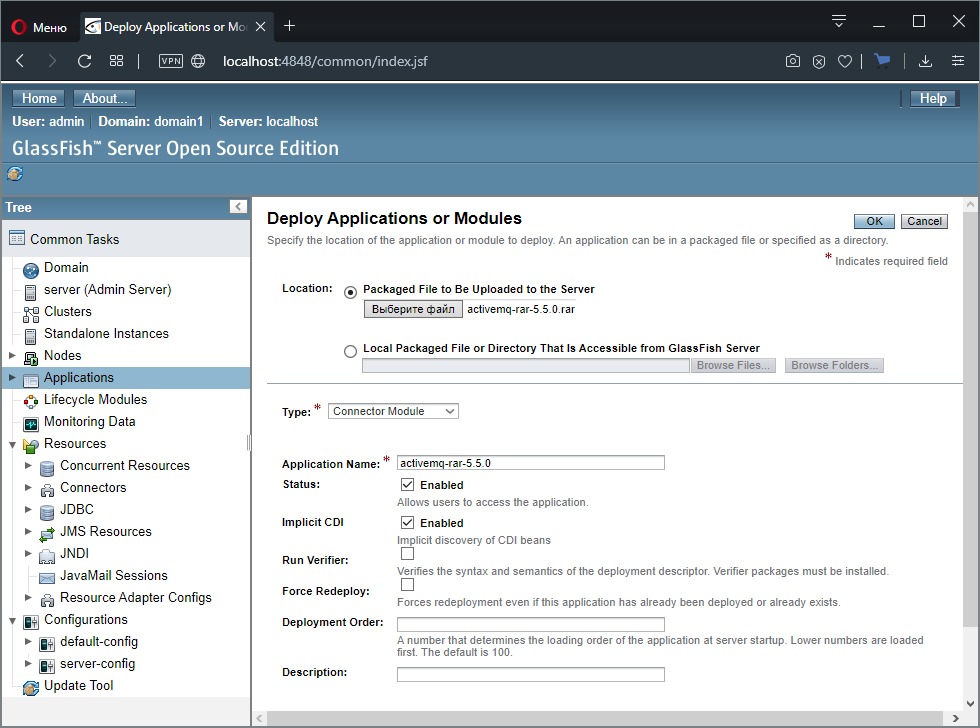
Трифонов Д.А.

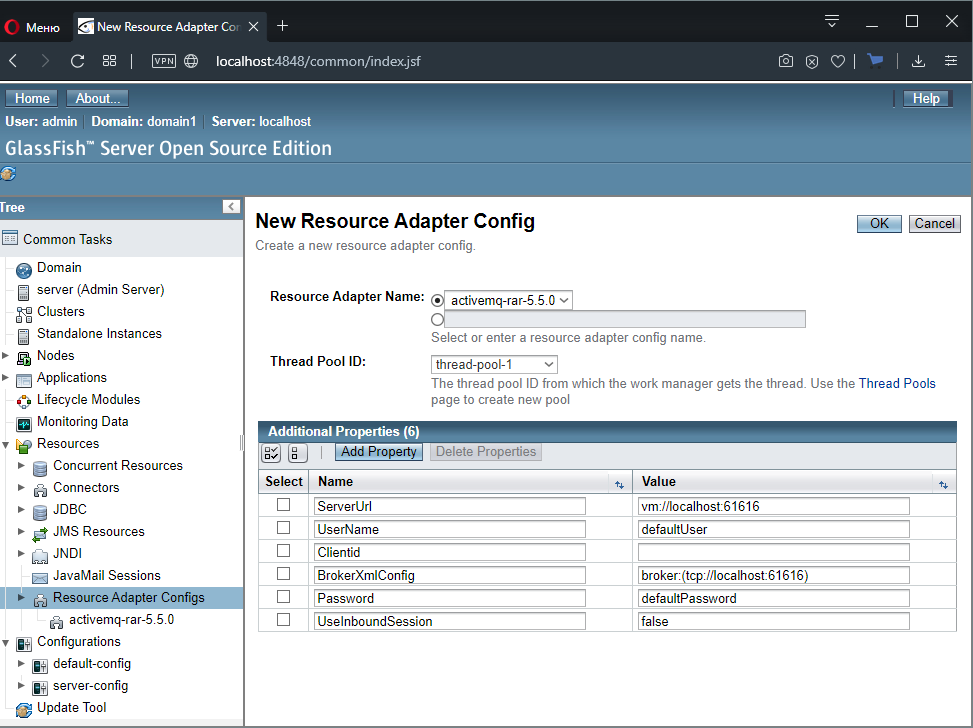
Владимир, 2020

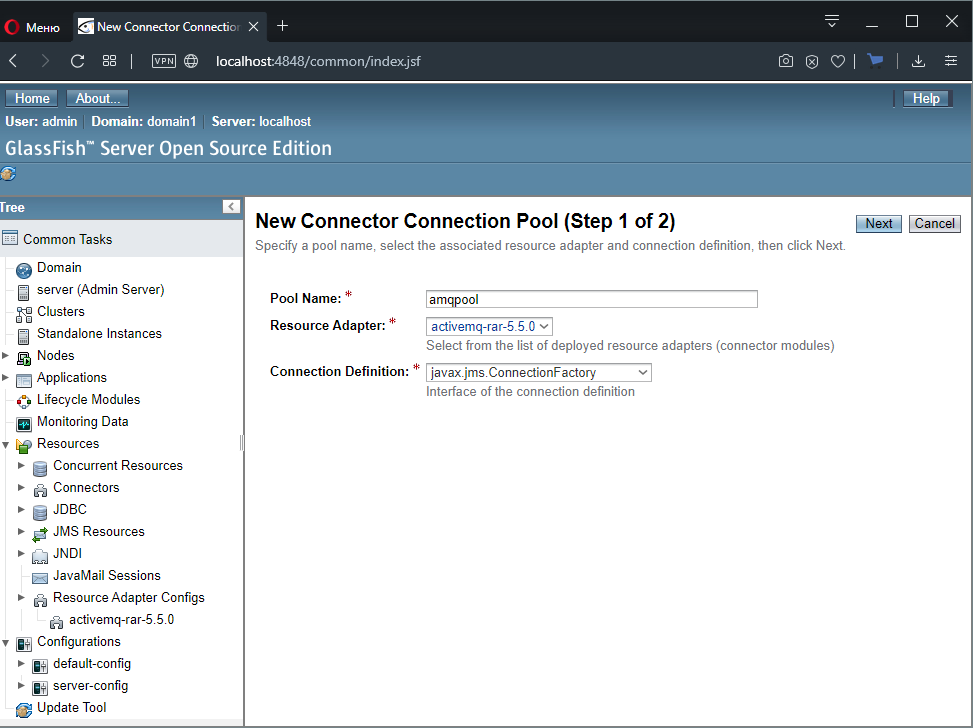
ЦЕЛЬ РАБОТЫ

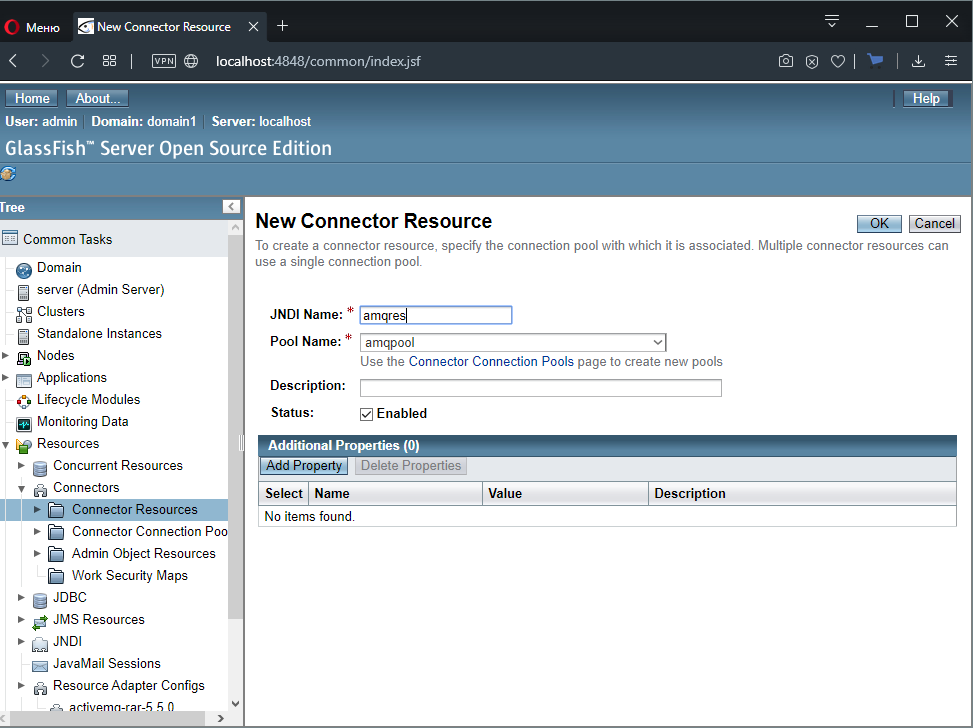
Получить навыки работы с системами обмена сообщениями и реализации приложений на платформе Java, взаимодействующих путем обмена сообщениями.

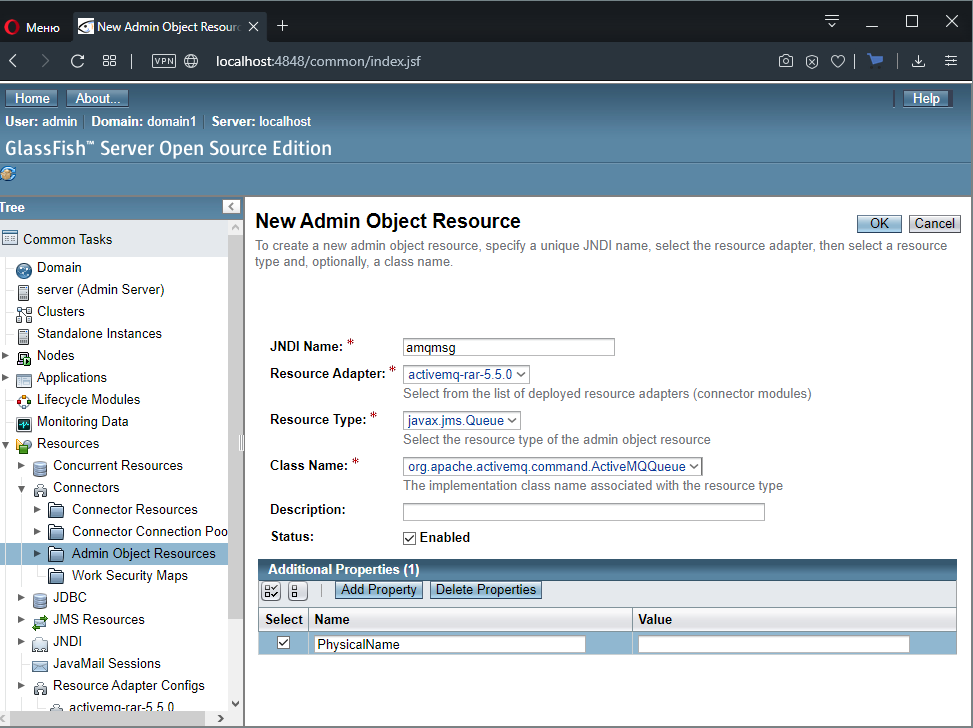
ВЫПОЛНЕНИЕ РАБОТЫ

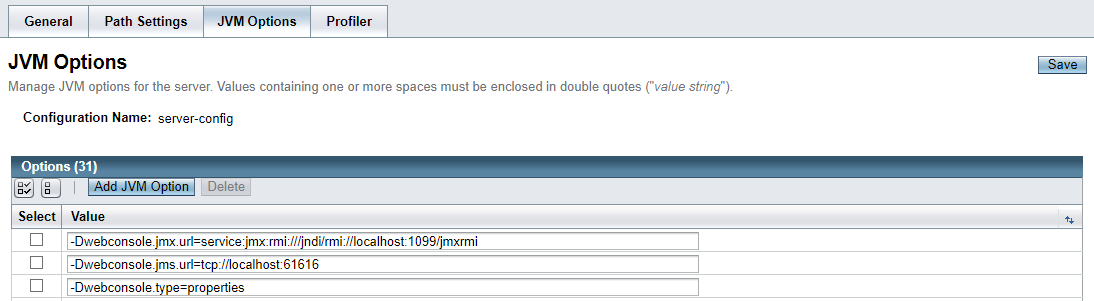


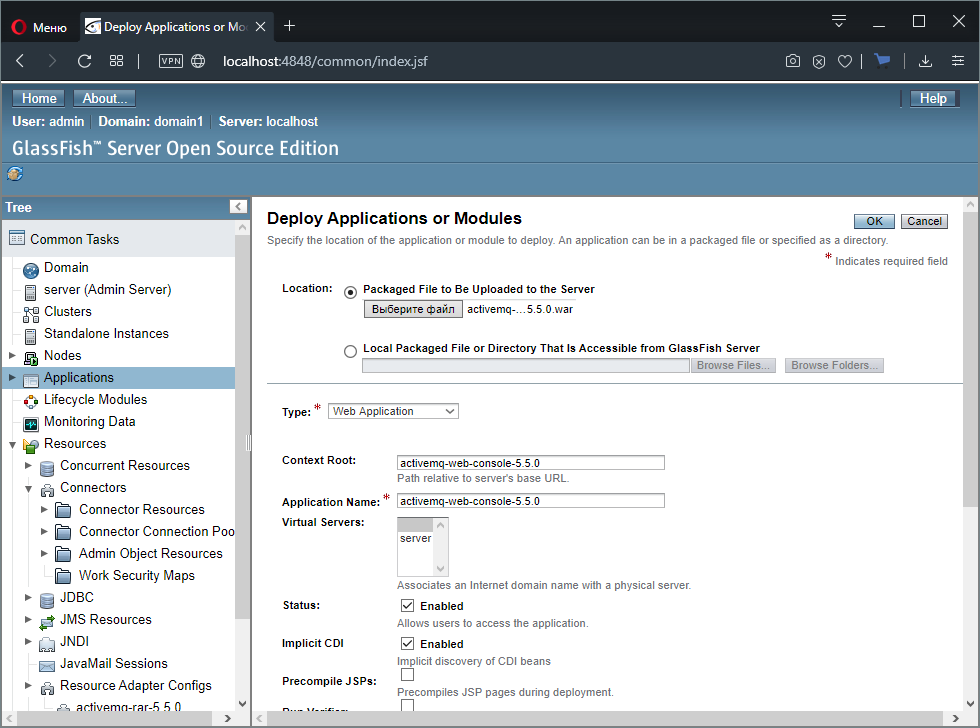


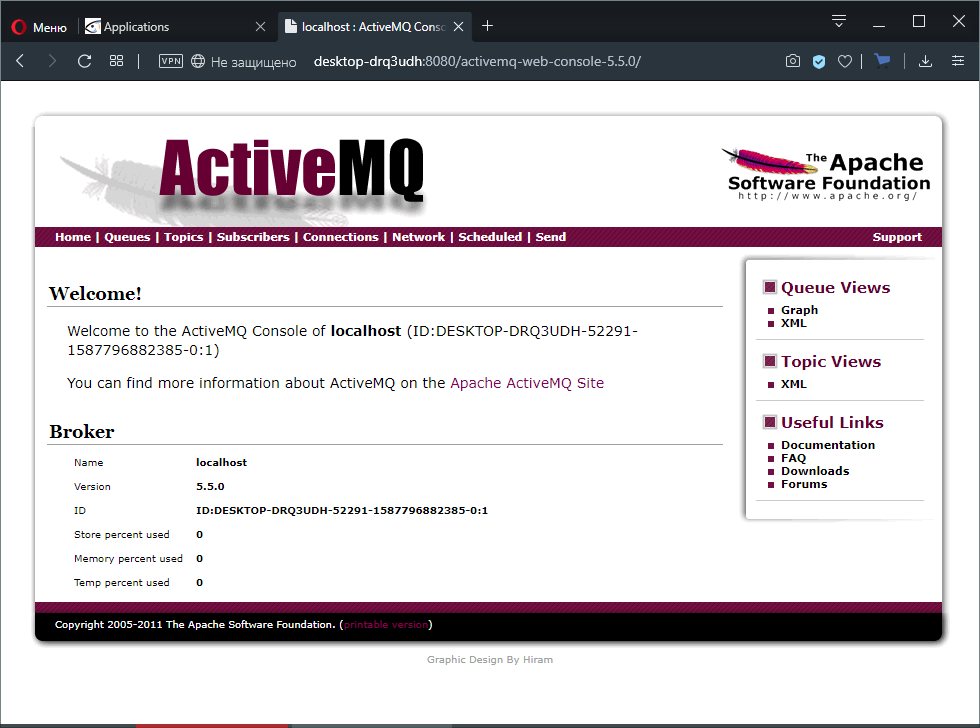


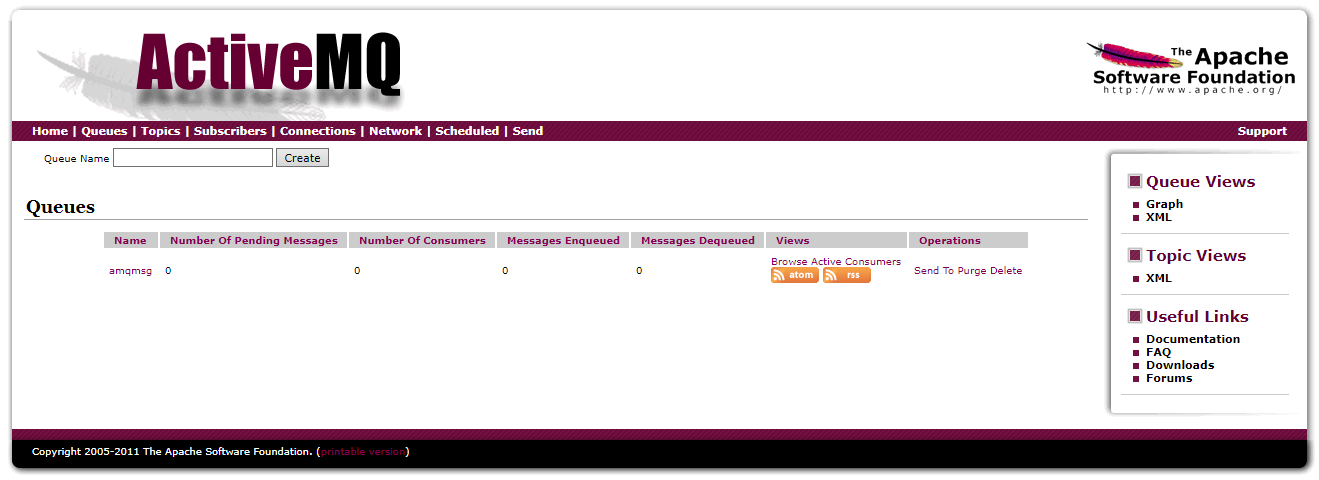


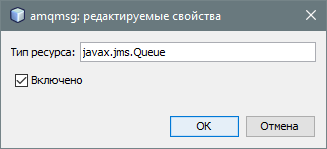












package simplsendamq;

import javax.jms.Connection;

import javax.jms.DeliveryMode;

import javax.jms.Destination;

import javax.jms.MessageProducer;

import javax.jms.Session;

import javax.jms.TextMessage;

import org.apache.activemq.ActiveMQConnection;

import org.apache.activemq.ActiveMQConnectionFactory;

public class SimpleSendAMQ {

private String user = ActiveMQConnection.DEFAULT\_USER;

private String password = ActiveMQConnection.DEFAULT\_PASSWORD;

private String url = ActiveMQConnection.DEFAULT\_BROKER\_URL;

private Destination destination;

public static void main(String[] args) {

SimpleSendAMQ mySend = new SimpleSendAMQ();

mySend.sendMessage();

}

void sendMessage() {

try {

Connection connection = null;

ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory(user, password, url);

connection = connectionFactory.createConnection();

connection.start();

Session session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

destination = session.createQueue("amqmsg");

MessageProducer producer = session.createProducer(destination);

producer.setDeliveryMode(DeliveryMode.NON\_PERSISTENT);

TextMessage message = session.createTextMessage("THIS IS THE TEST MESSAGE !");

producer.send(message);

connection.close();

} catch (Exception ex) {

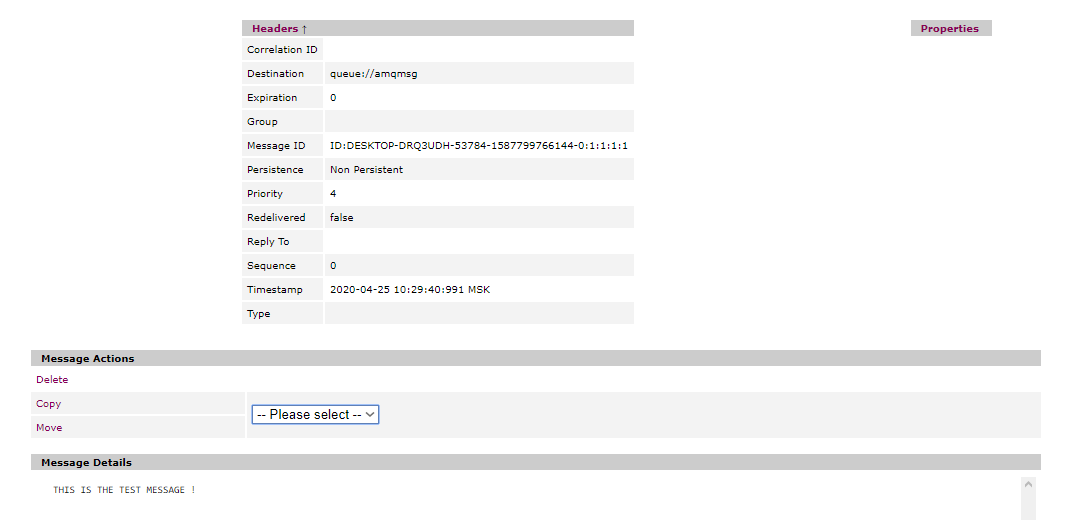
System.out.println(ex.getMessage());

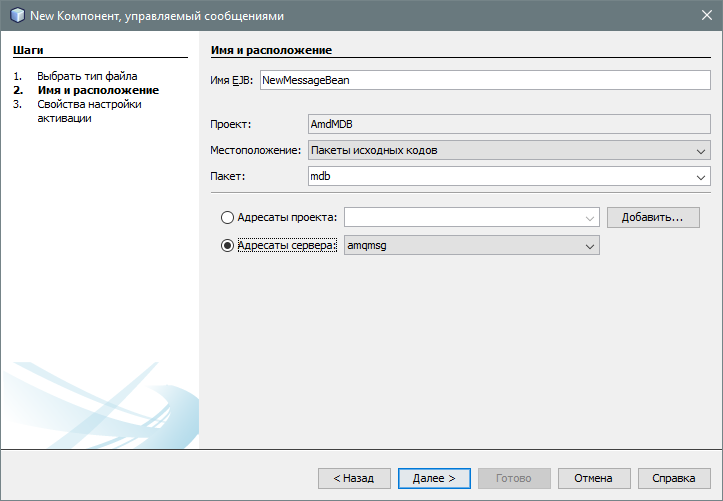
}

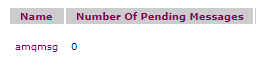
}

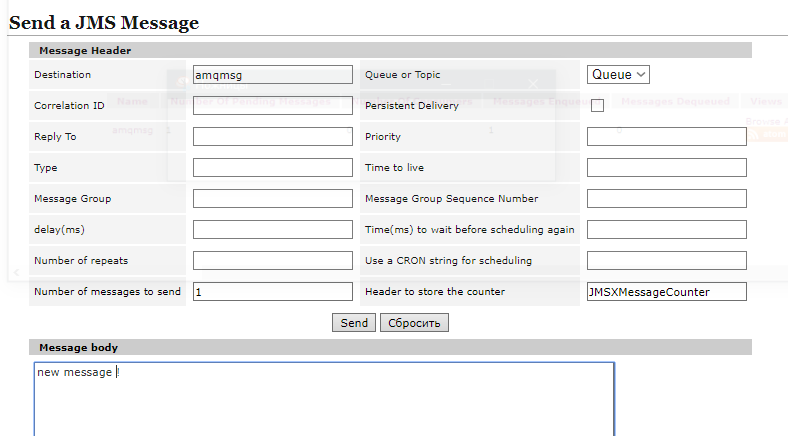
}











Создание веб-приложения для отправки сообщений и получения сообщений из очереди.

package bean;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.inject.Named;

import javax.enterprise.context.ApplicationScoped;

import javax.jms.Connection;

import javax.jms.DeliveryMode;

import javax.jms.Destination;

import javax.jms.JMSException;

import javax.jms.Message;

import javax.jms.MessageConsumer;

import javax.jms.MessageProducer;

import javax.jms.Session;

import javax.jms.TextMessage;

import org.apache.activemq.ActiveMQConnection;

import org.apache.activemq.ActiveMQConnectionFactory;

/\*\*

 \*

 \* @author Лера

 \*/

@Named(value = "newManagedBeanJMS")

@ApplicationScoped

public class NewManagedBeanJMS {

    private final String user = ActiveMQConnection.DEFAULT\_USER;

    private final String password = ActiveMQConnection.DEFAULT\_PASSWORD;

    private final String url = ActiveMQConnection.DEFAULT\_BROKER\_URL;

    private Destination destination;

    public void sendMessage(String textMsg) {

        try {

            Connection connection = null;

            ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory(user, password, url);

            connection = connectionFactory.createConnection();

            connection.start();

            Session session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

            destination = session.createQueue("amqmsg");

            MessageProducer producer = session.createProducer(destination);

            producer.setDeliveryMode(DeliveryMode.NON\_PERSISTENT);

            TextMessage message = session.createTextMessage(textMsg);

            producer.send(message);

            connection.close();

        } catch (JMSException ex) {

           Logger.getLogger(this.getClass().getName()).log(Level.WARNING, "Cannot close session", ex);

        }

    }

    public String receiveMessage(){

        Connection connection = null;

        String msgStr = "";

        try {

            ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory(user, password, url);

            connection = connectionFactory.createConnection();

            Session session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

            destination = session.createQueue("amqmsg");

            MessageConsumer consumer = session.createConsumer(destination);

            connection.start();

            Message msg;

            msg = consumer.receive(60);

            if (msg==null) return "В очереди нет сообщений";

            TextMessage textMsg = (TextMessage)msg;

            msgStr = msg.getJMSMessageID() + " ТЕКСТ: " + textMsg.getText();

            connection.close();

        }catch (JMSException ex) {

            Logger.getLogger(this.getClass().getName()).log(Level.WARNING, "Cannot close session", ex);

        }

        return msgStr;

    }

    public NewManagedBeanJMS() {

    }

}

Контроллер:

package controller;

import bean.NewManagedBeanJMS;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

 \*

 \* @author Лера

 \*/

public class JMSServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        NewManagedBeanJMS jms = new NewManagedBeanJMS();

        String receiveMessage = jms.receiveMessage();

        request.setAttribute("msg", receiveMessage);

        RequestDispatcher requestDispatcher = request.getRequestDispatcher("index.jsp");

        requestDispatcher.forward(request, response);

    }

    @Override

    protected void doPost(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        String text = request.getParameter("text");

        NewManagedBeanJMS jms = new NewManagedBeanJMS();

        jms.sendMessage(text);

        RequestDispatcher requestDispatcher = request.getRequestDispatcher("index.jsp");

        requestDispatcher.forward(request, response);

    }

}

Jsp-cтраница:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

       <form action="JMSServlet" method="POST">

            Текст сообщения:

            <textarea name="text"></textarea>

            <button>Отправить</button>

        </form>

        <form action="JMSServlet" method="GET">

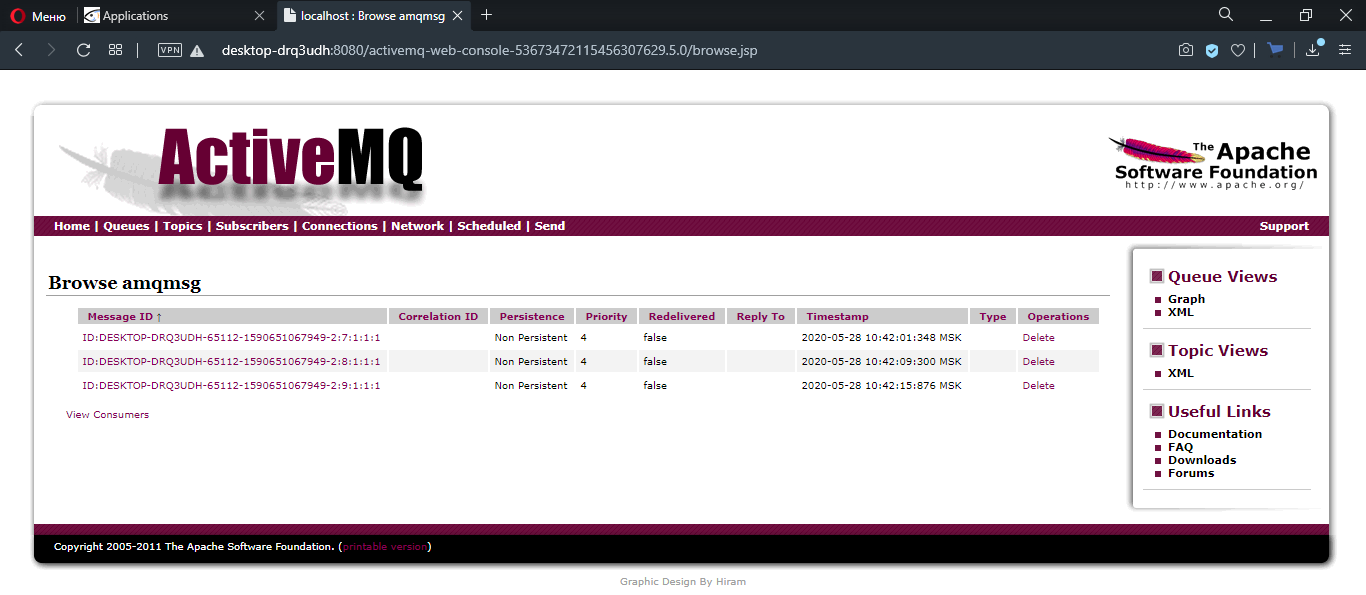
            <button>Получить сообщение из очереди</button>

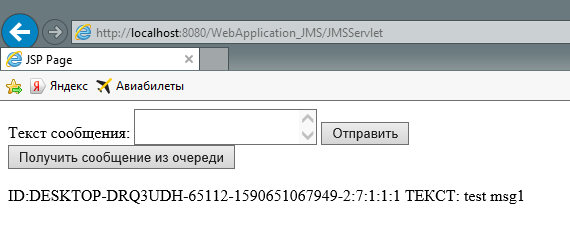
        </form><br>

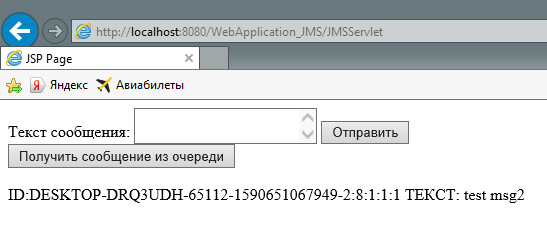
        <c:out value="${msg}"/>

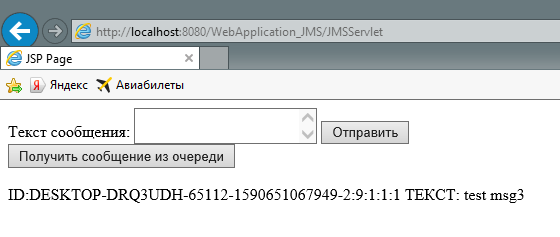
    </body>

</html>









ВЫВОД

В ходе выполнения лабораторной работы были получены навыки работы с системами обмена сообщениями и реализации приложений на платформе Java, взаимодействующих путем обмена сообщениями.