

1. Prin ce diferă cele două modele de mesagerie JMS p2p și pub-sub? Imaginați câte un use case adecvat.

P2P - În acest model, mesajele sunt trimise de un producător către o coadă specifică. Fiecare mesaj este procesat de un singur consumator. Un use case ar fi un sistem bancar unde tranzacțiile ar trebui să fie procesate secvențial.

Pub-Sub – Mesajele sunt publicate pe un topic la care se abonează consumatorii. Fiecare mesaj publicat poate fi consumat de oricare consumator. Un use case ar fi actualizarea stocului în timp real.

2. Ce reprezintă o Sesiune JMS? Dar un Topic?

Sesiune JMS = un context în care se desfășoară trimiterea și primirea mesajelor. O sesiune poate crea cozi, topicuri, mesaje și poate gestiona tranzacții.

Topic = un canal de distribuție la care mesajele sunt publicate de producători și la care consumatorii se abonează pentru a primi mesajele. (a le consuma)

3. Ce proprietăți pot fi asignate mesajelor JMS- enunțați și explicați utilitatea pentru minim 3 proprietăți?
 - JMSDeliveryMode – determină dacă mesajul trebuie să fie persistent
 - JMSPriority – stabilește prioritatea unui mesaj pe o scală de la 0 la 9
 - JMSExpiration – setează timpul după care mesajul expiră dacă nu este livrat
 - JMSTimestamp
 - JMSDestination
 - JMSType
4. Descrieți succint arhitectura JGroups.

JGroups este o bibliotecă Java pentru comunicarea în grupuri care permite dezvoltarea de protocoale de rețea. Arhitectura sa cuprinde mai multe straturi printre care stratul de transport, protocoale și API de aplicație.

5. Cum se pot identifica nodurile de procesare (hosturile) prezente la un moment dat în clusterul creat cu ajutorul tehnologiei JGroups?

Nodurile într-un cluster JGroups pot fi identificate și gestionate folosind diverse metode:

- Vizualizarea stării grupului: folosind instrumente și comenzi JGroups pentru a obține lista nodurilor active.
 - Ascultarea evenimentelor: interceptarea evenimentelor de adăugare sau eliminare a nodurilor din cluster.
6. Comparați din perspectiva complexității codului cele două implementări JMS, respectiv JGroups pentru Aplicația chat

JMS: Ar putea fi mai simplu de implementat pentru un chat de bază folosind modelul pub-sub, dar gestionarea conexiunilor și a sesiunilor poate complica codul.

JGroups: Oferă un nivel mai înalt de abstracție pentru comunicarea în grup, simplificând astfel implementarea unui chat. Totuși, configurarea inițială a stivei de protocoale poate fi mai complexă.

7. Pentru ilustrarea unui mecanism simplu de replicare ce utilizează tehnologia Jgroups, testați și documentați (UML class diagram) mini-sistemul de cacheuri replicate