

Send the finished task by e-mail to your teaching assistant. Deadlines for the submission are on the web page of the course: <http://d3s.mff.cuni.cz/teaching/middleware/>.

The assignment describes a message-based communication protocol that implements a simple trading system. Understanding the assignment requires no special knowledge.

Prerequisites

The chosen messaging standard is JMS (Java Message Service), using the ActiveMQ 5.14 implementation. The following knowledge is needed for the implementation:

- Basic methods provided by the JMS specification, and their application (creating objects of type `Connection`, `Session`, `Queue`, `Topic` etc.).
- The method of implementing message producers and consumers (`MessageProducer` and `MessageConsumer` objects).
- The method of creating and parsing messages of different types (`TextMessage`, `ObjectMessage`, `MapMessage` etc.), knowledge of criteria for choosing an appropriate message type.
- Executing the JMS infrastructure (service provider).

Simple examples using JMS will be available.

Assignment details

<http://d3s.mff.cuni.cz/teaching/middleware/files/as3.zip>

Implement an online trading application. Each running instance communicates with other running instances and also with the Bank.

Application must provide following functionality:

1. After its startup, create random unique names of goods, that the user (seller) provides, together with random price of each item. This list should be sent as a message via the “Offers” topic to the other running instances.

The application should also send the Bank a message requesting account creation. The bank will respond with a message containing the account number.

2. Present to the users (in any trivial form) a list of goods and prices, offered by the other running instances. This list should be updated, whenever a new message from the “Offers” topic arrives, and any items no longer offered by the sending instance should be removed from the list.
3. Allow the user to choose (again, using any trivial method) goods to buy from another instance. The sale is realized as follows:
 - The buyer sends a message to the seller, containing the name of the requested item and his (the buyer’s) account number.
 - The seller sends a message to the buyer, containing his (the seller’s) account number, or indicating a refusal to sell (the item was already sold etc.).
 - The buyer sends a message to the Bank, requesting transfer of appropriate amount of money from his account to the seller’s account.
 - The bank sends a message to the seller, notifying him of the money transfer from the buyer’s account.

- The seller removes the item from its list and sends a message to the buyer, confirming the finished sale.

Note: The buyer does not have to offer the item bought for sale.

4. Consider the weaknesses of the communication protocols described above. Extend, document and implement a more robust variant of the protocol (reusing the provided parts is allowed) that will include at least:
 - Keeping proper account balances (newly created accounts can have fixed balance for simplicity) and support for account balance queries.
 - More robust sell/buy protocol, considering the buyer's account balance.
 - More robust sell/buy protocol, considering that the buyer may transfer less money than the actual price of the required item (assume the price is fixed).
 - Better availability of the goods offered by running instances to newly connected instances.

The messages used to implement the functionality may contain further information that might be needed but not mentioned above. You are to choose the appropriate message types. (For example, the provided Bank implementation uses `TextMessage` and `MapMessage`, you are free to choose different types for the communication between the client instances.)

The communication is significantly simplified in 1-3 (e.g. the Bank does not consider account balance), the implementation should however behave reasonably when possible – refuse to sell an item already requested by a different buyer, etc.

An implementation of the Bank and some parts of the solution will be provided.