

Only the best
got in ;-)

Online presentations soon available at <http://wiki.javapolis.com>



JAVAPOLIS



JavaPolis 2004



Mock objects as testing method

- Eric Seynaeve
- Instructor
ABIS



TRAINING & CONSULTING



Outline

- Introduction JUnit
- Sample application
- Introduction Mock objects
- Testing with Mock objects
- Conclusions





Where does the need for Mock objects come from and how to use them.





Introduction JUnit

- Unit testing framework (www.junit.org)
- Automates testing and some reporting
- Most important class: **TestCase**
- failures vs. errors





Introduction JUnit: TestCase

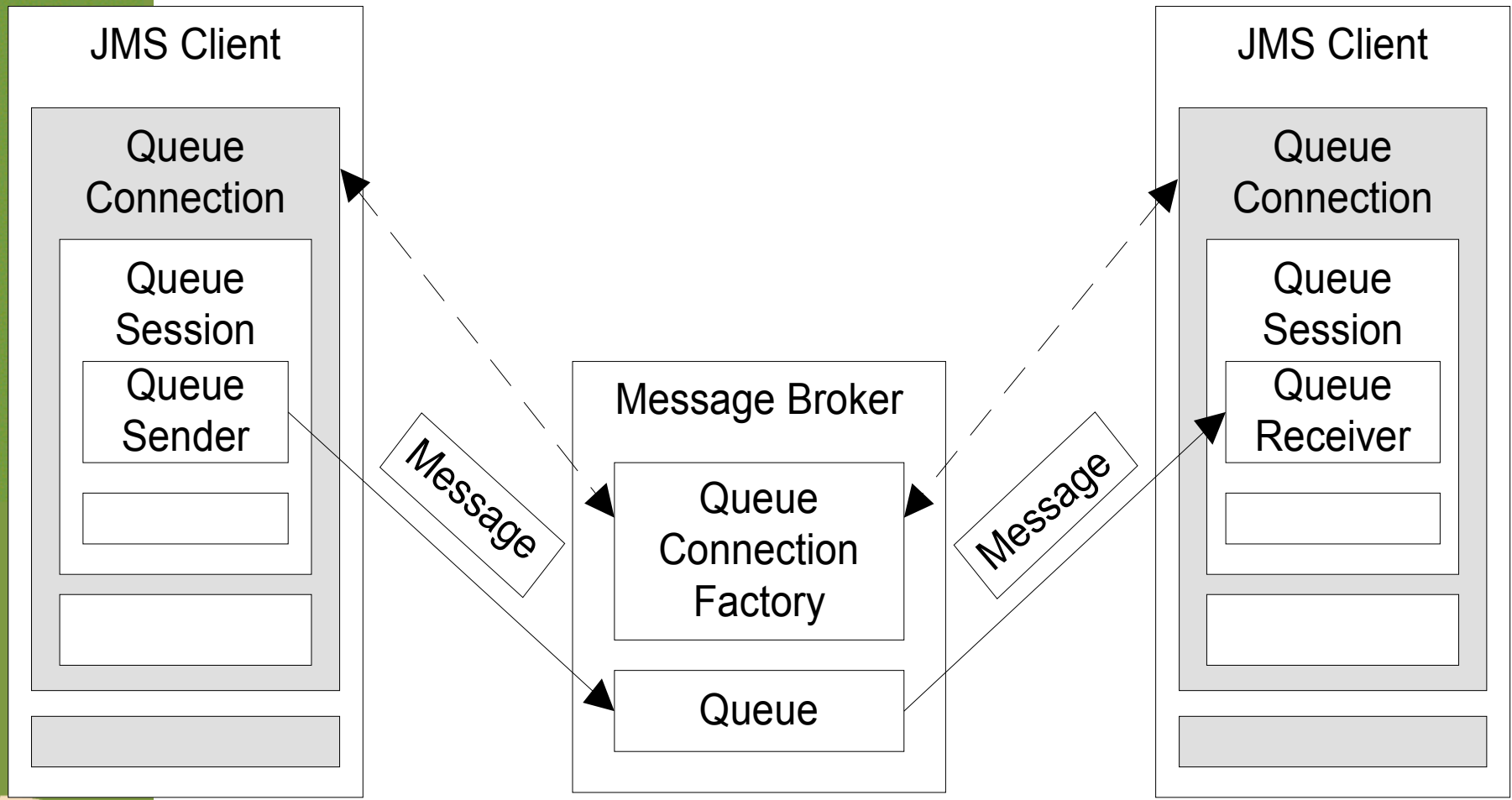
Most important `TestCase` methods

- `setUp()`
- `testXxx()`
- `fail()`





Quick JMS overview





Sample application

- Simple JMS client application
 - Put a text message on the queue
 - Use a utility class to encapsulate JNDI code
 - Do not forget to close the connection
 - Chain low-level exceptions (JNDI and JMS)





Sample application: code

```
public class SampleApplication {  
  
    ...  
  
    private void sendMessage() throws SampleApplicationException {  
        try {  
            QueueConnection qc = QueueUtils.getConnection();  
            QueueSession qsess = qc.createQueueSession(false, Session.AUTO_ACKNOWLEDGE);  
  
            Queue q = QueueUtils.getQueue("jms/abisQueue");  
            QueueSender qsend = qsess.createSender(q);  
  
            TextMessage msg = qsess.createTextMessage();  
            msg.setText(message);  
            qsend.send(msg);  
            qc.close();  
        } catch (CreationException e) {  
            throw new SampleApplicationException("Running sendMessage()", e);  
        } catch (JMSEException e) {  
            throw new SampleApplicationException("Running sendMessage()", e);  
        }  
    }  
}
```





Problems for testing

- MQ infrastructure needs to be running
- How to force exceptions from infrastructure code?
- How to check if the connection is closed?
- ...





Introduction to Mock objects

- Complements JUnit
- Replace infrastructure code with Mock objects
- Can involve refactoring of tested code!





Testing with Mock objects: step 1

- Write Mock objects
 - Start from skeleton code
 - Contain certain expectations





Testing with Mock objects: step 2

- Inject Mock objects into tested code
 - 2 common strategies:
 - Pass as argument to tested method
 - Use a Factory Method





First test

Demo

Test conversion of exceptions





What was done?

- Refactored code to include Factory Method
- Created skeleton implementations of JMS interfaces
- Tested code





Second test



Demo
Test close connection





What was done?

- Added expectation to Mock object
- Tested code





Third test



Demo

Test close connection
when exception is thrown





What was done?

- Combined above test methods
- Tested code
- Corrected bug :-)





Making things generic

- Some code is generic
 - Extract it into reusable classes
- Don't code yourself
 - Include JMS, JDBC, ... Mock objects

www.mockobjects.com





Testing with mock objects

Demo

Rewriting tests using Mock objects
from mockobjects.com





General test cookbook with Mock objects

- Set up Mock infrastructure (in `setup()`)
- Set up expectations
- Run code to be tested
- Verify expectations





Conclusions

Mock objects:

- Test calls to infrastructure
- Can uncover design problems
- Isolate code bugs from infrastructure bugs





Q&A



Code examples:
eseynaeve@abis.be



JavaPolis 2004



abis

TRAINING & CONSULTING

<http://www.abis.be>

Thank you

