▸ Pour Commencer

▸ Week 0: Introduction to Network and Service Management

▸ Week 1: Key Concepts with SNMP

▸ Week 2: Monitoring with Nagios

▾ **Week 3: Instrumentation with JMX**

Overview of the Content

Lecture 1: Key Concepts and Architecture
Lesson_Quiz

Lecture 2: Basic Instrumentation
Lesson_Quiz

Lecture 3: Support Services
Lesson_Quiz

Practical Exercise 1: JMX and JConsole
Practical_Exercise_Quiz

Practical Exercise 2: Standard MBean
Practical_Exercise_Quiz

Practical Exercise 3: Dynamic MBean
Practical_Exercise_Quiz

Practical Exercise 4: Adding a Notification
Practical_Exercise_Quiz

Evaluations
Week_Evaluation Echéance le avril 10, 2022 at 22:00 UTC

Aidez-nous à améliorer ce MOOC

▸ Week 4: Next-Generation Management Protocols

▸ Votre avis nous intéresse

EVALUATIONS W3_EV (20 points possibles)

Question W3.EV.1: the JMX Agent Services include: (NA=2)

☑ A MBean Server ✓

□ A MBean Persistance Service

□ A MBean Versioning Service

☑ A Naming Service ✓

EXPLANATION

Versioning and Persistance are not provided by a JMX Agent. If such services are required, they have to be implemented by the management solution designer.

Question W3.EV.2: in the JMX Notification Model, the filtering is done by: (NA=1)

○ The Notification

○ The Broadcaster

● The Emitter ✓

○ The Listener

EXPLANATION

The consumer (listener) specifies the filter but the filtering itself is done at the Emitter which implements the NotificationBroadcaster interface.

Question W3.EV.3: which of the following deployment models enable application life cycle management? (NA=2)

□ Component

☑ Driver ✓

☑ Daemon ✓

□ Delegation

EXPLANATION

Only the Driver and Daemon deployment models ease the life-cycle management of Java-based applications instrumented in JMX.

Question W3.EV.4: JMX has built-in: (NA=2)

☑ Mbeans attributes monitoring ✓

□ Distributed (multi-JVM) Mbeans management

□ Persistent Mbeans storage

☑ At runtime management interface building ✓

EXPLANATION

A simple monitoring service for attributes is provided and Dynamic + Model Mbeans allow to build the management interface of a managed resource at runtime.

Question W3.EV.5: let MbeanA and MBeanB be two standard MBeans with defined Management interfaces A and B. Assume MBeanA is a subclass of MbeanB. Select the statements which are true, given this configuration: (NA=2)

☑ MBeanA always exposes the attributes and operations from MBeanA ✓

□ MbeanA always exposes the management attributes and operations from MBeanB

□ MBeanB always exposes the attributes and operations from MBeanA

☑ MbeanA exposes the attributes and operations of MBeanA and MbeanB if the management interface of B is defined as a subinterface from A ✓

EXPLANATION

See lesson 2, slide 23 describing the inheritance rules.

Question W3.EV.6: the Dynamic MBean Management Operations are defined: (NA=2)

☑ at runtime ✓

☑ in an MBeanInfo object ✓

□ at compile time

□ In a Java interface

□ In a configuration file

EXPLANATION

The MBeanServer delegates the build of the management interface to the Mbean itself through the creation of an MBeanInfo object at runtime, more precisely at the MBean registration time.

Question W3.EV.7: JMX relations are: (NA=2)

□ dynamically defined

☑ MBeans defined ✓

☑ User defined ✓

Question W3.EV.8: Remote Monitoring in JMX is by default: (NA=2)

☑ provided through a standard Java connector ✓

□ provided through an SNMP gateway

□ provided though a CMIS management stack

☑ Available through an HTML adaptor ✓

Question W3.EV.9: the JMX Default Agent supports: (NA=2)

☑ A notification subscription service ✓

□ TCP network connection abstractions modeled as relationships

☑ Remote management ✓

□ Memory dumping

Question W3.EV.10: One JVM can host several MBeanServers: (NA=1)

● True ✓

○ False

Question W3.EV.11: the instruction ManagementFactory.getPlatformMBeanServer() returns: (NA=1)

○ An instance of a MBean server

● An instance of the platform MBean server ✓

○ A server instance

Question W3.EV.12: a standard MBean should implement the JMX class StandardMBean: (NA=1)

○ Yes

● No ✓

Question W3.EV.13: a standard MBean is composed of an MBean interface and a class: (NA=1)

● Yes ✓

○ No

Question W3.EV.14: by convention, the class name of a MBean interface, takes the name of the class that implements it, with the suffix MBean added. (NA=1)

● Yes ✓

○ No

Question W3.EV.15: which of these declarations defines a readable attribute of a standard MBean interface? (NA=1)

● public int getNumberOfCities() ✓

○ public void getNumberOfCities()

○ None of them

Question W3.EV.16: which of these declarations defines a writable attribute of a standard MBean interface? (NA=1)

○ public int setSizeOfThreadsPool(int size)

● public void setSizeOfThreadsPool(int size) ✓

○ None of them

Question W3.EV.17: a dynamic MBean requires an MBean interface? (NA=1)

○ Yes

● No ✓

Question W3.EV.18: which exception is generated when an attribute is not found in a dynamic MBean? (NA=3)

☑ MBeanException ✓

□ FileNotFoundException

☑ ReflectionException ✓

☑ AttributeNotFoundException ✓

□ None of them

Question W3.EV.19: to generate a notification, the MBean should extend: (NA=1)

● NotificationBroadcasterSupport ✓

○ NotificationEmitter

○ None of them

Question W3.EV.20: to send a notification, you need to construct an instance of the class (NA=2)

☑ javax.management.Notification ✓

☑ javax.management.Notification or a subclass (such as AttributeChangedNotification) ✓

□ javax.management.MBeanNotificationInfo