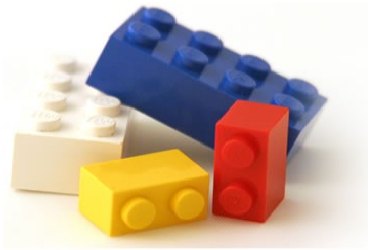
CS5233

Components – Models and Engineering

- Komponententechnologien -

Master of Science (Informatik)



Java Management Extensions: JMX

http://download.oracle.com/javase/tutorial/jmx/index.html

JMX Java Management Extensions

API for management and monitoring of applications, services, ... and JVM

Provides standard interface between a resources and its manager

Usage:

- view or change configuration
- observe behavior
- notify of state changes

JMX usually is supported by application servers or servlet containers

Relevant Specifications

JSR 3, *Java Management Extensions (JMX) Specification, version 1.4*

JSR 160, *JMX Remote API*

JSR 255, *Java Management Extensions (JMX) Specification, version 2.0*

Status

Package `javax.management`

Introduced in Java 5

Considerable extension in Java 6 (based on JSR 3 / version 1.4)

No modification in Java 7

JSR 255 postponed to Java 8

MBeans and Manageable Resources

Manageable Resource

entity – hardware or software – (JVM, servlet, EJB, …) that is managed

MBean

represents a manageable resource

provides management interface for the resource

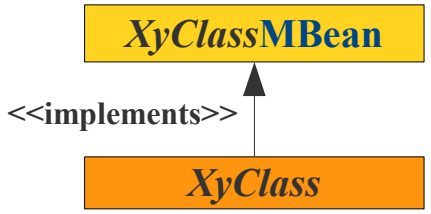
often is identified with the managed resource

may be

- the managed resource itself if the managed resource is a Java object
- or a wrapper around legacy code
- or a proxy
- or ...

Standard MBean

there are several types of MBeans, the simplest is the standard MBean



- **Standard MBean**
  - implements its own **custom MBean** interface
  - **MBean interface**
    - must have a name that ends in `MBean`
    - must have a signature that exposes attributes and operations to the management application
  - **MBean immplementation**
    - must have a name that equals the interface's name without Mbean
    - must implement
      - operational code
      - management code according to the interface

the other kinds of MBeans are:

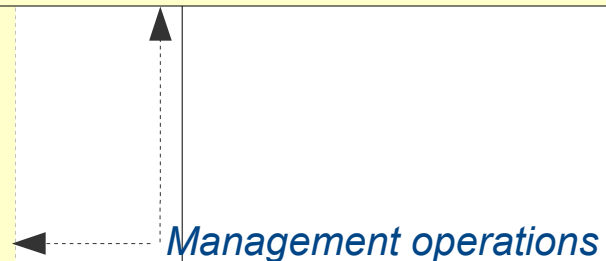
- **MXBeans**
- **Dynamic MBeans**
- **Model MBeans**
- **Open MBeans**

JMX / Standard MBean

Example: A manageable bounded Stack

```
public class BoundedStack implements BoundedStackMBean {
  int length = 10;
  int top    = -1;
  private int[] a = new int[length];

  public void push(int x) { a[++top] = x; }
  public int pop() { return a[(top--)]; }

  @Override
  public void reset() { a = new int[length]; top = -1; }

  @Override
  public int getLength() { return length; }

  @Override
  public void resize(int length) {
      this.length = length;
      this.top = -1;
      this.a = new int[length];
  }

  @Override
  public List<Integer> getState() {
     List<Integer> res = new ArrayList<Integer>(top+1);
      for (int i=0; i<=top; i++) { res.add(a[i]); }
      return res;
  }
}
```

```
public interface BoundedStackMBean
{
   void reset();
   int getLength();
   void resize(int length);
   List<Integer> getState();
}
```



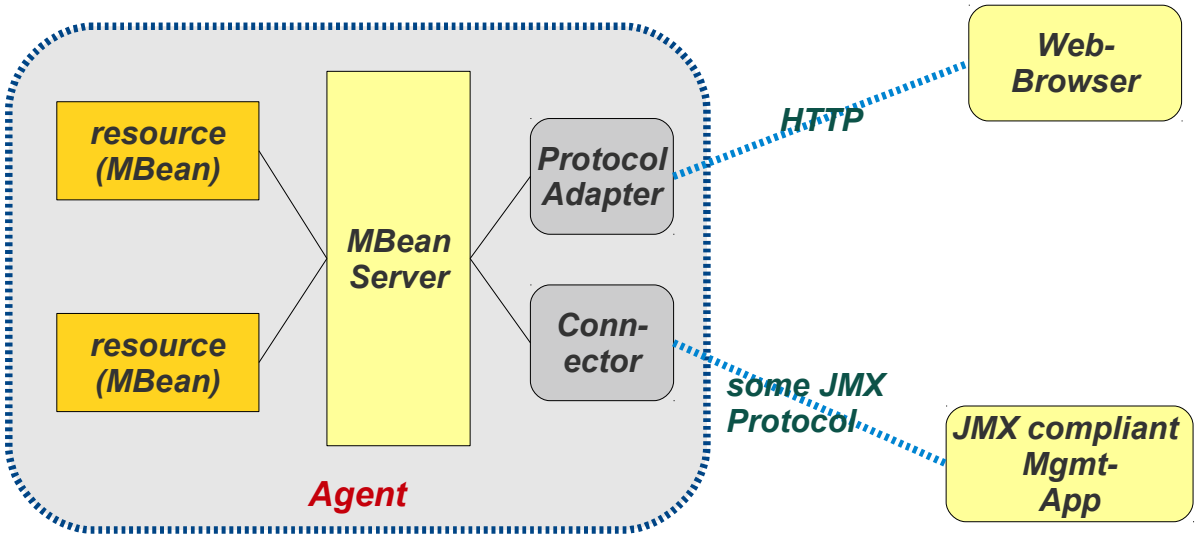
JMX Agent

An JMX agent **manages** one or more resources

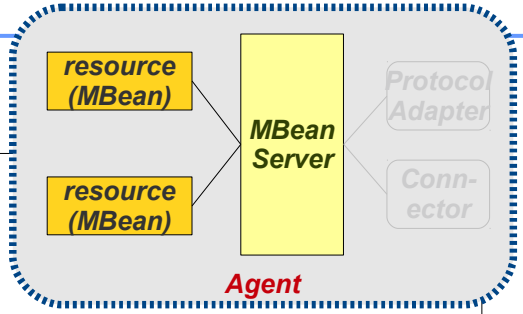
It connects resources to a management application

An agent consists of

- **MBean server** (one)
- **MBeans** as resources (one or more)
- **Connector** or **Protocol Adapter** (one or more)

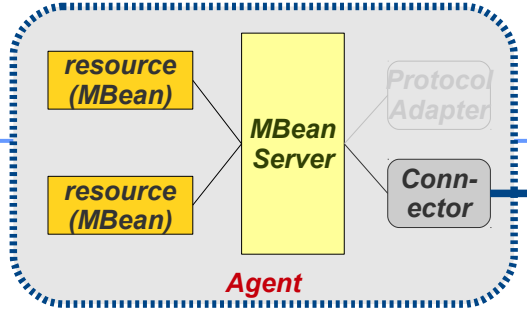


JMX Agent example (1)



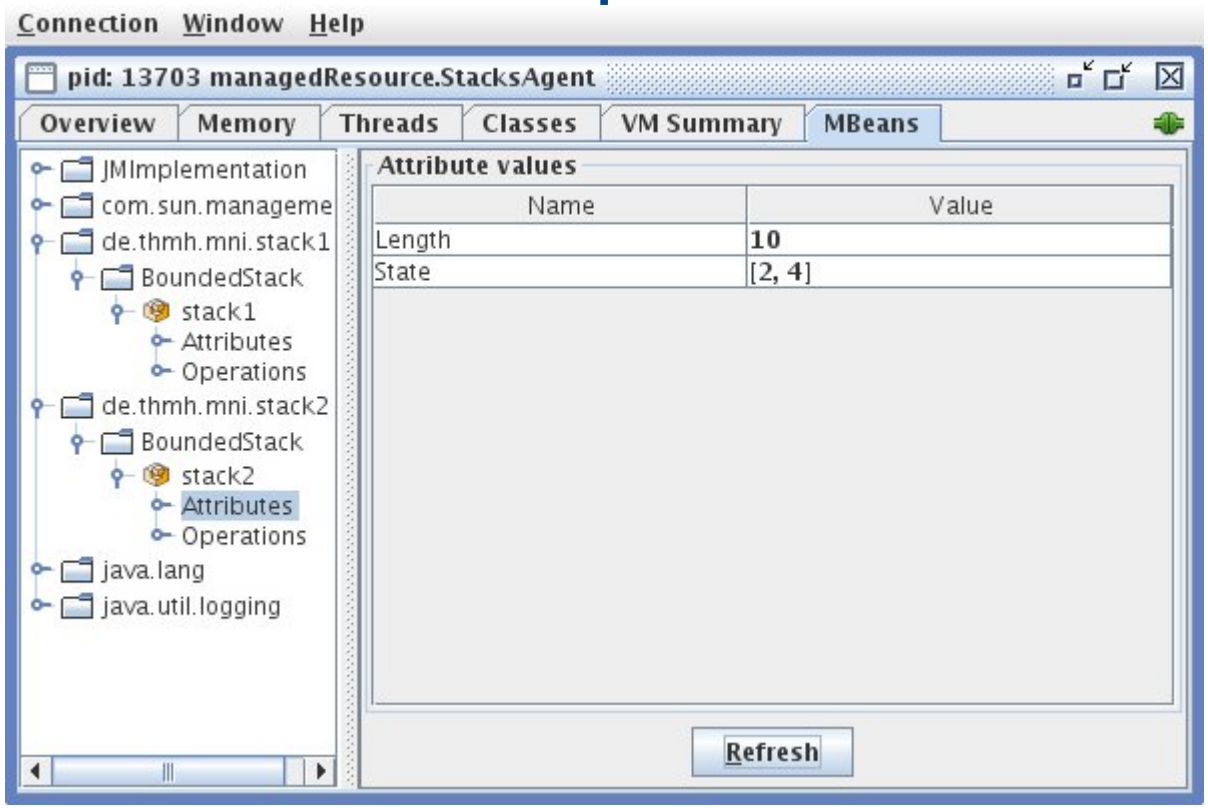

```
import java.lang.management.ManagementFactory;
import javax.management.InstanceAlreadyExistsException;
import javax.management.MBeanRegistrationException;
import javax.management.MBeanServer;
import javax.management.MalformedObjectNameException;
import javax.management.NotCompliantMBeanException;
import javax.management.ObjectName;

public class StacksAgent {
  public static void main(String[] args) throws MalformedObjectNameException,
                    NullPointerException,
                    InstanceAlreadyExistsException, MbeanRegistrationException,
                    NotCompliantMBeanException, InterruptedException {

    MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();

    ObjectName name1 = new ObjectName("de.thmh.mni.stack1:type=BoundedStack,name=stack1");
    ObjectName name2 = new ObjectName("de.thmh.mni.stack2:type=BoundedStack,name=stack2");

    BoundedStack stackMBean1 = new BoundedStack();
    BoundedStack stackMBean2 = new BoundedStack();

    mbs.registerMBean(stackMBean1, name1);
    mbs.registerMBean(stackMBean2, name2);


    }
}
```

```
while(true) {
  Thread.sleep(1000);  stackMBean1.push(1);
  Thread.sleep(1000);  stackMBean2.push(2);
  Thread.sleep(1000);  stackMBean1.push(3);
  Thread.sleep(1000);  stackMBean2.push(4);
  Thread.sleep(1000);  stackMBean2.pop();
  Thread.sleep(1000);  stackMBean1.pop();
  Thread.sleep(1000);  stackMBean2.pop();
  Thread.sleep(1000);  stackMBean1.pop();
}
```



JMX Agent example (2)

jconsole: command line management application connects to the agent (on local machine)



JMX compliant Mgmt-App

`<jdk-7>/bin/jconsole`

Dynamic MBean

Dynamic and static MBeans publish their interfaces. Whereas for static MBeans this is done "automatically" (JMX libraries do it using refection), for dynamic MBeans you have to code the publishing.

Dynamic MBeans

- **are used if manage resources with interfaces not known in advance and have to be determined at run-time**
- **must implement `javax.management.DynamicMBean`**
- **operational code**
- **management code**
- **publishing that provides access to operational code**

JMX / Dynamic MBean

Example (1)

```
import java.util.ArrayList;
import java.util.List;

import javax.management.Attribute;
import javax.management.AttributeList;
import javax.management.AttributeNotFoundException;
import javax.management.DynamicMBean;
import javax.management.InvalidAttributeValueException;
import javax.management.MBeanException;
import javax.management.MBeanInfo;
import javax.management.MBeanOperationInfo;
import javax.management.MBeanParameterInfo;
import javax.management.ReflectionException;

public class BoundedStack implements DynamicMBean {
  int length            = 10;
  int top   = -1;
  private int[] a = new int[length];

  // Operations

  public void push(int x) {
    if (top >= length) { throw new IllegalStateException(); }
    a[++top] = x;
  }

  public int pop() {
    if (top < 0) { throw new IllegalStateException(); }
    return a[(top--)];
}
```

Example (2)

```
// Management
public void reset() {
  a = new int[length];
  top = -1;
}

public int getLength() { return length; }

public void resize(int length) {
  if (length < 0) { throw new IllegalArgumentException(); }
  this.length = length;
  this.top = -1;
  this.a = new int[length];
}


public List<Integer> getState() {
  List<Integer> res = new ArrayList<Integer>(top+1);
  for (int i=0; i<=top; i++) { res.add(a[i]); }
  return res;
}
```

JMX / Dynamic MBean

Example (3a)

```
//Publishing code
@Override
public MBeanInfo getMBeanInfo() {
   MBeanOperationInfo[] opers = {
        new MBeanOperationInfo(
          "reset",                  //name of method
          "Reset the stack",        //description
           null,                    //parameters
           "void",                  //return type
           MbeanOperationInfo.ACTION //has an effect but does not return any information
         ),
        new MBeanOperationInfo(
            "getLength",            //name of method
            "Get stack length",     //description
            null,                   //parameters
            "int",                  //return type
            MBeanOperationInfo.INFO //INFO: no effect returns information
        ),
        new MBeanOperationInfo(
            "resize",               //name of method
            "Resize the stack",     //description
            new MBeanParameterInfo[] {//parameters
               new MBeanParameterInfo(
                        "length",               //parameter name
                        "int",                  //parameter type
                        "The new size")         //description
            },
            "int",                  //return type
            MBeanOperationInfo.INFO //INFO: no effect returns information
        ),
        new MBeanOperationInfo(
            "getState",             //name of method
            "Get stack state",      //description
            null,                   //parameters
            "List<Integer>",        //return type
            MBeanOperationInfo.INFO //INFO: no effect returns information
        )
   };
```

Example (3b)

```
//public MBeanInfo getMBeanInfo() { continued

   return new MBeanInfo(
         this.getClass().getName(), //The name of the Java class of the MBean described by this MBeanInfo.
            "BoundedStack MBean",   //A human readable description of the MBean
                null,               //The list of exposed attributes of the MBean
                null,               //The list of public constructors of the MBean
                opers,              //The list of operations of the MBean
                null);              //The list of notifications emitted
}
```

Example (3c)

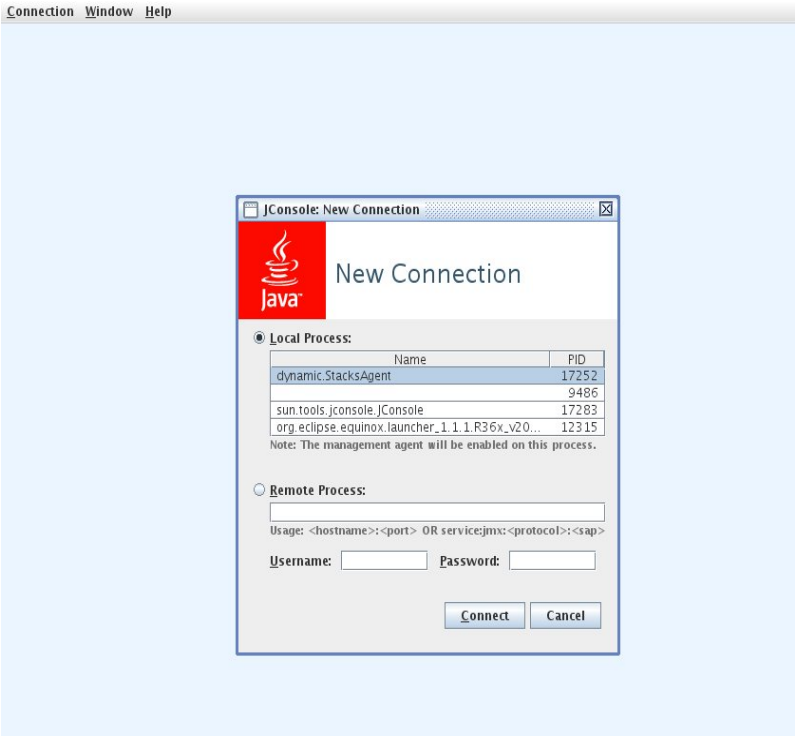
```
@Override
public Object invoke(String name, Object[] args, String[] signature) throws MBeanException, ReflectionException {
  if (name.equals("reset") &&
      (args == null || args.length == 0) && (signature == null || signature.length == 0)) {
      try {
         reset();
         return null;
      } catch (Exception e) { throw new MbeanException(e); }
  }

  if (name.equals("getLength") &&
      (args == null || args.length == 0) && (signature == null || signature.length == 0)) {
     try {
         return getLength();
     } catch (Exception e) { throw new MbeanException(e); }
  }

  if (name.equals("resize") &&
      (args.length == 1) && (signature.length == 1 && signature[0].equals("int"))) {
     try {
         resize((Integer)args[0]);
     } catch (Exception e) { throw new MbeanException(e); }
  }

  if (name.equals("getState") &&
      (args == null || args.length == 0) && (signature == null || signature.length == 0)){
     try {
         return getState();
     } catch (Exception e) { throw new MbeanException(e); }
  }

  throw new ReflectionException(new NoSuchMethodException(name));
}
```

Example (3d)

```
    @Override
    public Object getAttribute(String arg0) throws AttributeNotFoundException,
                                                    MBeanException, ReflectionException {
            throw new AttributeNotFoundException();
    }

    @Override
    public AttributeList getAttributes(String[] arg0) {
            return new AttributeList();
    }

    @Override
    public void setAttribute(Attribute arg0) throws AttributeNotFoundException,
                                    InvalidAttributeValueException, MBeanException, ReflectionException {
            throw new AttributeNotFoundException();
    }

    @Override
    public AttributeList setAttributes(AttributeList arg0) {
            return new AttributeList();
    }

}
```

JMX / Dynamic MBean

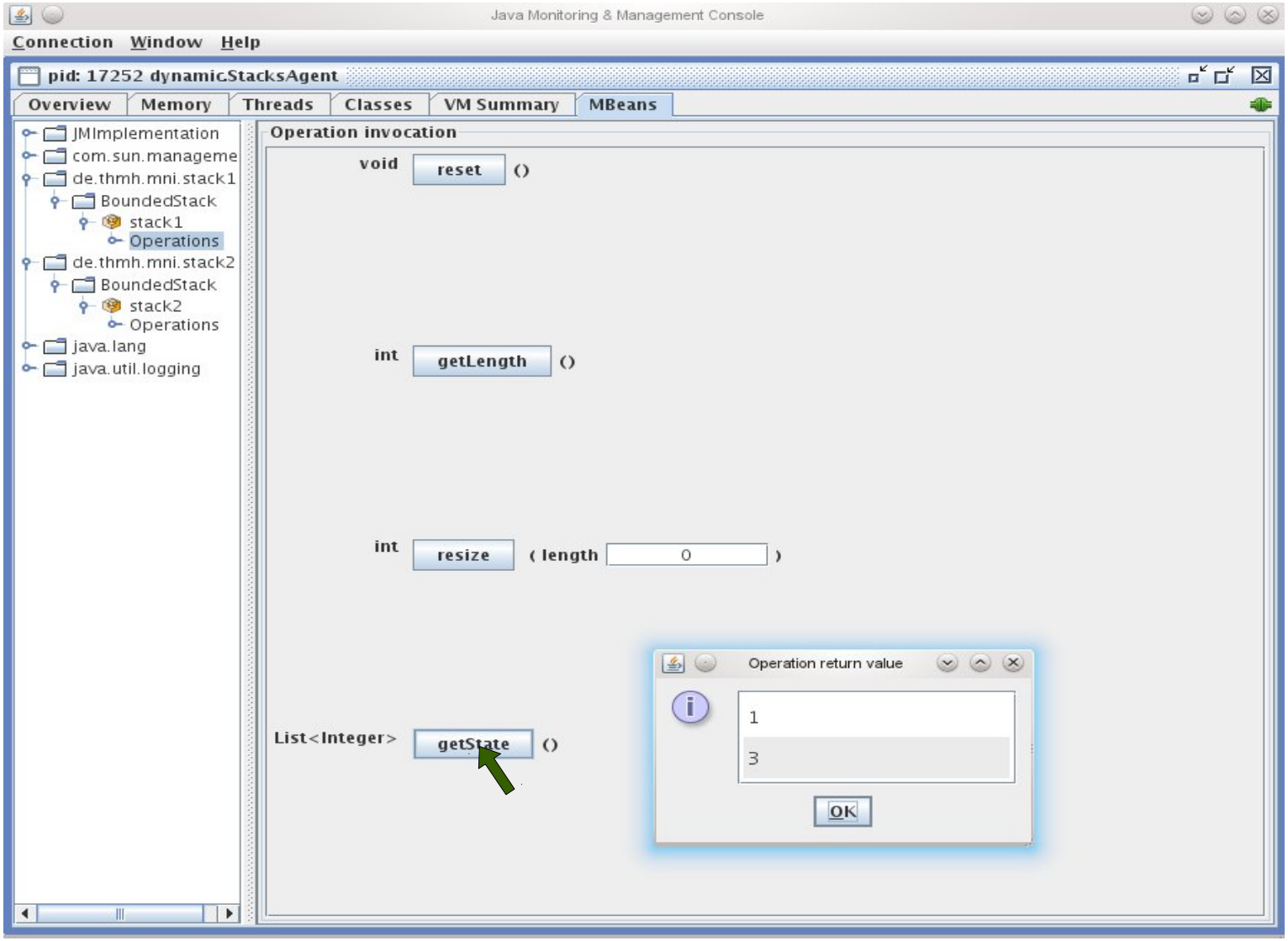
Example (4a)




```
.../jdk1.6.0_23/bin/jconsole
```

JMX / Dynamic MBean

Example (4b)



… and many things more

see:

JSR-000003 *JavaTM Management Extensions (JMXTM)* Evaluation 1.4 Maintenence Release 3
http://www.jcp.org/en/jsr/detail?id=3