

Java Management Extension

JMX

- JMX Java Management Extension расширение для управления приложениями
- Какие выгоды предлагает JMX?
 - простота реализации JMX основана на понятии сервера и объектов управления MBeans
 - сервер выступает как агент управления, объекты инструментарий
 - масштабируемость службы агентов JMX независимы, и могут быть встроены в агент JMX
 - гибкость JMX позволяет расширять концепцию в будущем, создавать любые решения в будущем
 - управляемость JMX API представляет сервисы для работы в распределенной среде
- Пример. реализация JMX Local [java02/jmx/jmxsc](#)
 - процедура запустить два Terminal
 - Terminal 1 запустить приложение check.cmd
 - Terminal 2 запустить jconsole check1.cmd
 - проверка JConsole выбрать процесс с портом 10999
 - MBean найти SimpleAgent
 - запустить sayHello(), изменить message, запустить sayHello()
- Пример. реализация JMX Remote [java02/jmx/jmxr](#)
 - процедура запустить Main() прямо из IDE
 - проверка JConsole
 - RemoteProcess вставить строку "service:jmx:rmi:///jndi/rmi://localhost:9999/server "
 - MBean найти SimpleAgent
 - запустить sayHello(), изменить message, запустить sayHello()
- Пример. реализация JMX Remote manual [java02/jmx/jmxr](#)
 - процедура встать на каталог project (parent of src)
 - создать три окна Terminal
 - Treminal1 src/java02/jmxr/check.cmd регистрация порта
 - Treminal2 src/java02/jmxr/check1.cmd компиляция и запуск агента
 - Treminal3 src/java02/jmxr/check2.cmd запуск JConsole
 - проверка JConsole
 - RemoteProcess вставить строку "service:jmx:rmi:///jndi/rmi://localhost:9999/server "
 - MBean найти SimpleAgent
 - запустить sayHello(), изменить message, запустить sayHello()
- Пример. реализация JMX Html [java02/jmx/jmxh](#)
 - процедура запустить Main() прямо из IDE
 - проверка FireFox
 - <http://localhost:8000>
 - MBean найти SimpleAgent >> name=hellothere
 - запустить sayHello(), изменить message, запустить sayHello()

- Что еще умеет JMX кроме дистанционного управления?
 - просмотр и изменение конфигурации приложения
 - сбор и публикация статистики о работе приложения
 - извещение о нештатных ситуациях или изменении состояния приложения
 - event notification извещение о событиях изменения атрибута
 - monitor service уведомления о зарегистрированных событиях
 - timer service уведомления по таймеру
 - m-let service создание и регистрация экземпляров MBean Server
- Что такое MBean?
 - MBean объекты реализуют JMX интерфейс, который включает
 - величины которые доступны
 - методы которые могут быть вызваны
 - извещения которые могут быть посланы
 - конструкторы
- Какие типы MBeans существуют?
 - существует четыре типа MBeans standard, dynamic, open и model
 - Standard интерфейс определяется набором методов
 - Dynamic специализированный интерфейс, доступный во время исполнения
 - Open это dynamic MBean, использует только основные типы данных
 - Model это dynamic MBean, полностью конфигурируется и может показать свой интерфейс
 - Notification Model базовая модель основана на Event для передачи данных между агентами JMX
 - MBean Metadata Classes классы, которые используются для создания MBean интерфейсов
- Пример. реализация Model MBean [java02/jmxmodel/jmxm](#)
 - применение создать класс приложения
 - зарегистрировать сервер
 - создать массив параметров
 - зарегистрировать массив приложение, массив параметров
 - запустить сервер
 - запустить JConsole
- Что такое MBean Server?
 - сервер это реестр объектов MBeans, которые используются для управления
 - предоставляет стандартный интерфейс к объектам MBean
 - регистрация обязательно для любого MBean к которому нужен доступ
 - регистрировать MBean может другой MBean, агент или удаленное приложение
 - при регистрации объекту MBean назначается уникальное имя

Ключи JVM

- Ключи JVM
 - `-Xbootclasspath` ключ загрузки базовых классов
 - `-classpath` ключ загрузки классов пользователя