Учебный центр компьютерного обучения «Специалист» при МГТУ им. Н.Э.Баумана

Виды bean-компонентов

Типы Enterprise Bean`ов

Учебный центр компьютерного обучения «СПЕЦИАЛИСТ» при МГТУ им. Н.Э.Баумана

Цели занятия:

- Ознакомиться с различными типами бинов
- Определить область применения каждого типа бинов
- Изучить альтернативы, плюсы и минусы бинов
- Закрепить знания на практике

Bean`ы бывают:

- **Session beans**
  - Stateful Session Beans
  - Stateless Session Beans
  - Singleton Session Beans
    - Startup Bean
- **Message driven beans (MDBs)**
- **Entity beans (заменены на Java Persistence API)**

Минусы EJB:

- Нет поддержки на
  - TomCat
  - Jetty
  - Resin
  - Jrun
- Плохая обратная совместимость

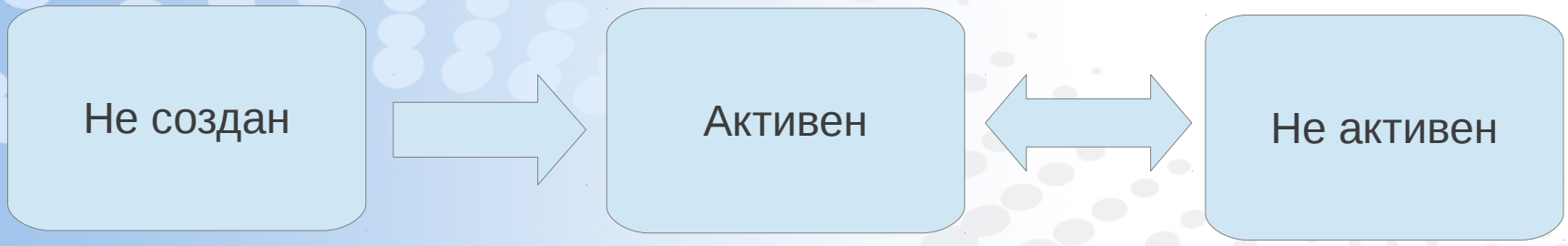
Плюсы EJB:

- Энкапсуляция бизнес-логики
- Удаленный доступ
- Поддержка основных вендоров
  - Jboss
  - GlassFish
  - OAS
  - WebLogic
  - WebSphere
- Масштабируемость

Сравнение Session Beans и Message-Driven Beans

Session Beans	Message-Driven Beans
Вызываться напрямую через интерфейс	Вызываются по событию из очереди сообщений
Синхронный вызов (клиент ожидает завершения вызова)	Асинхронный вызов
Возвращает клиенту объект, как обычный вызов метода.	Не возвращает клиенту значения, но может вызвать callback
Основан на интерфейсах	Класс не разделяется на интерфейс и реализацию

Жизненный цикл



Состояния при создании:
- Создание
- Dependency Injection
- Вызов метода PostConstruct
- InitMethod

Состояния при удалении:
- PreDestroy (@PreDestroy)

Жизненный цикл. Аннотации

@PostConstruct
@PostActivate
@PrePassivate
@Remove
@PreDestroy

Interface Session Bean

- Local
- Remote

Бин реализует оба интерфейса.

Интерфесы могут наследоваться друг от друга. Обычно Local расширяет Remote.

Над интерфейсами ставится аннотация @Local или @Remote соответственно

При передаче данных по TCP информация сериализуется.

=>

Объект передаваемых по сети должен имплементировать интерфейс Serializable

Класс, имплементирующий интерфейсы

- Помечается аннотацией Statless или Statful
  - Аннотации можно передать параметр mappedName= для возможности обращения к бину по JNDI.
  - Если параметр не передать, то бин можно использовать локально через аннотацию @EJB
  - Доступ к локальной функциональности может быть наследован или делегирован

Клиенты RMI (1/2)

- Обращение к бину происходит через JNDI
- Контекст содержащий подключение обычно располагается в контексте в ClassPATH
- Настройки для GlassFish
  - org.omg.CORBA.ORBInitialHost=<ipAddress>
- Настройки для Jboss
  - java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
  - java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
  - java.naming.provider.url=<ipAddress>:1099

Клиенты RMI (2/2)

- Клиенты общаются с бином как с обычным объектом POJO
- Для получения Бина:
  - InitialContext context = new InitialContext();
  - InterfaceName bean = (InterfaceName)context.lookup("JNDI-Name");

JNDI по умолчанию

- Если не задать mappedName
  - Jboss
    - <BeanImpl_name>/remote
  - Glassfish
    - <package>.<Interface_name>

MDB

- Server
  - Класс бина
    - реализует интерфейс MessageListener
    - Переопределяет onMessage
    - Отмечается аннотацией @MessageDriven
    - Конфигурация очереди передаётся с AS
- Client
  - Находит ConnectionFactory и Queue в контексте
  - Создает Session и MessageProducer
  - Определяет тип сообщения

Типы сообщений

- Текстовое сообщение (String)
- Объектное сообщение (Serializable)
- Свойства (String + примитив)
- Бинарное сообщение (поток байт)
- Потоковое сообщение (передача примитивов)
- Для определения типа используется instanceof

Типы сообщений

- Текстовое сообщение (String)
- Объектное сообщение (Serializable)
- Свойства (String + примитив)
- Бинарное сообщение (поток байт)
- Потоковое сообщение (передача примитивов)
- Для определения типа используется instanceof

Типы сообщений

- Текстовое сообщение (String)
- Объектное сообщение (Serializable)
- Свойства (String + примитив)
- Бинарное сообщение (поток байт)
- Потоковое сообщение (передача примитивов)
- Для определения типа используется instanceof

Типы сообщений

- Текстовое сообщение (String)
- Объектное сообщение (Serializable)
- Свойства (String + примитив)
- Бинарное сообщение (поток байт)
- Потоковое сообщение (передача примитивов)
- Для определения типа используется instanceof