

## **Принципы обработки ошибок EJB**

С точки зрения платформы Java EE исключения, выбрасываемые методами EJB-компонентов, делятся на 2 большие группы:

- прикладные (Application Exception)
- системные (System Exceptions)

При их обработке контейнер выполняет разные действия.

Прикладные исключения – это исключения, которые сообщают клиенту о специфических для предметной области проблемах (например, некорректное значение входного параметра), преодоление которых зависит от клиента. Прикладное исключение передается клиенту точно в том виде, в каком оно выбрасывается методом компонента.

Классы прикладных исключений определяются разработчиком. При этом класс исключения не должен расширять `java.rmi.RemoteException`.

Прикладные исключения могут быть как контролируемые, так и неконтролируемые. В первом случае они должны быть указаны в предложении `throws` бизнес-метода компонента (как в бизнес-интерфейсе, так и в классе компонента). Во втором случае класс исключения должен быть помечен аннотацией `@ApplicationException` (данную аннотацию можно применять и к контролируемым исключениям).

При возникновении прикладного исключения контейнер не выполняет автоматический откат транзакции, позволяя клиенту преодолеть ошибочную ситуацию. В некоторых случаях для гарантирования целостности данных следует пометить транзакцию для отката при выбросе прикладного исключения. Это можно сделать явно (вызовом метода `EJBContext.setRollbackOnly()` перед выбросом прикладного исключения) или неявно, пометив класс прикладного исключения как исключение, вызывающее откат транзакции:

```
@ApplicationException(rollback=true)
public class RollingBackAppException { ... }
```

К системным исключениям относятся неконтролируемые исключения (не помеченные как прикладные) и исключения, расширяющие `java.rmi.RemoteException`.

Компонентные приложения выбрасывают системные исключения в случаях, когда происходят ошибки, не зависящие от клиента, и преодолеть которые невозможно. Например: невозможность получить соединение с БД или найти имя в JNDI-контексте, исключение `RemoteException` при удаленном вызове, ошибки при работе виртуальной машины JVM (потомки `java.lang.Error`) и прочие.

При возникновении системного исключения компонент должен выбросить некоторое неприкладное исключение в соответствии со следующими рекомендациями:

- 1) если при исполнении метода возникло неконтролируемое исключение, то необходимо просто передать его контейнеру (`propagate to the container`) – компонент не обязан обрабатывать подобное исключение;
- 2) если при выполнении операции метода возникло контролируемое исключение, которое не может быть преодолено, следует выбросить `javax.ejb.EJBException`, которое содержит в себе оригинальный объект исключения;
- 3) прочие ошибочные ситуации должны завершаться выбросом `javax.ejb.EJBException`.

`EJBException` – это неконтролируемое исключение, поэтому его не нужно объявлять в предложении `throws` бизнес-метода компонента.

Когда контейнер отлавливает системные исключения, выброшенные методами компонентов, он:

1. сохраняет информацию о нем в журнале;
2. выбрасывает `java.rmi.RemoteException` или его подкласс, если вызов был сделан через удаленный интерфейс или интерфейс веб-сервиса, или `javax.ejb.EJBException`, если вызов был сделан через локальный интерфейс.

При этом контейнер гарантирует следующее:

1. транзакция, в которой участвовал вызванный метод (выбросивший системное исключение) заканчивается откатом;
2. на том экземпляре компонента, на котором был вызван метод, не будет больше вызвано ни одного другого метода (экземпляр компонента будет удален).

Спецификация определяет действия контейнера по обработке системных исключений следующим образом. Действия контейнера и события на клиенте в случае декларативного управления транзакциями приведены в табл. 1. Данная таблица описывает реакцию на исключения, возникающие в 1) бизнес-методах компонентов, 2) обработчиках событий ЖЦ компонентов, 3) обработчиках вызовов бизнес-методов, 4) методах интерфейса `SessionSynchronization`, реализуемых компонентом.

При программном управлении транзакциями прикладное исключение выбрасывается повторно и клиент его получает. Системное исключение обрабатывается так:

- 1) фиксируется в журнале;
- 2) транзакция, начатая компонентом, помечается к откату;
- 3) экземпляр компонента удаляется;
- 4) выбрасывается `RemoteException` или `EJBException`.

При этом клиент получает соответствующее исключение (`RemoteException` или `EJBException`). Если клиент исполняется в контексте своей транзакции, то она может быть, а может и не быть помечена контейнером для отката.

Исключения из бизнес-методов и обработчиков в сессионных компонентах  
(декларативное управление транзакциями)

Условие	Выброшенное исключение	Действия контейнера	События на клиенте
Компонент используется в контексте клиентской транзакции	Прикладное	Повторный выброс исключения. Пометить транзакцию для отката, если прикладное исключение вызывает откат.	Получает оригинальное исключение. Транзакция может быть продолжена, если не была помечена для отката компонентом
	Системное	<ul style="list-style-type: none"> <li>- Фиксировать в журнале</li> <li>- Пометить транзакцию для отката</li> <li>- Удалить экземпляр компонента</li> <li>- Выбросить особое исключение</li> </ul>	Получает <code>javax.transaction.TransactionRollbackException</code> (при вызове через удаленное представление) или <code>javax.transaction.TransactionRollbackLocalException</code> . Продолжение транзакции бессмысленно
Компонент используется в контексте транзакции, начатой автоматически при вызове метода клиентом	Прикладное	Если компонент пометил транзакцию для отката или прикладное исключение вызывает откат, откатить и выбросить исключение повторно. Иначе подтвердить транзакцию и повторно выбросить исключение.	Получает оригинальное исключение. Если клиент используется в контексте своей транзакции, он может ее продолжать, так как она для отката не помечена.
	Системное	<ul style="list-style-type: none"> <li>- Фиксировать в журнале</li> <li>- Откатить транзакцию</li> <li>- Удалить экземпляр компонента</li> <li>- Выбросить исключение</li> </ul>	Получает <code>java.rmi.RemoteException</code> или <code>javax.ejb.EJBException</code> . Если клиент исполняется в контексте своей транзакции, то она может быть, а может и не быть помечена контейнером для отката.
Компонент исполняется вне контекста транзакции	Прикладное	Повторный выброс исключения.	Получает оригинальное исключение. Если клиент исполняется в контексте своей транзакции, он может ее продолжать, она для отката не помечена.
	Системное	<ul style="list-style-type: none"> <li>- Фиксировать в журнале</li> <li>- Удалить экземпляр компонента</li> <li>- Выбросить исключение</li> </ul>	Получает <code>java.rmi.RemoteException</code> или <code>javax.ejb.EJBException</code> . Если клиент исполняется в контексте своей транзакции, то она может быть, а может и не быть помечена контейнером для отката.