Practical – 7

Enterprise Java Beans (EJB)

AIM : Demonstrate the application of a simple stateful EJB.

Theory :

EJB is an acronym for enterprise java bean. It is a specification provided by Sun Microsystems to develop secured, robust and scalable distributed applications.

To run EJB application, you need an application server (EJB Container) such as Jboss, Glassfish, Weblogic, Websphere etc. It performs:

- life cycle management
- security
- transaction management
- object pooling

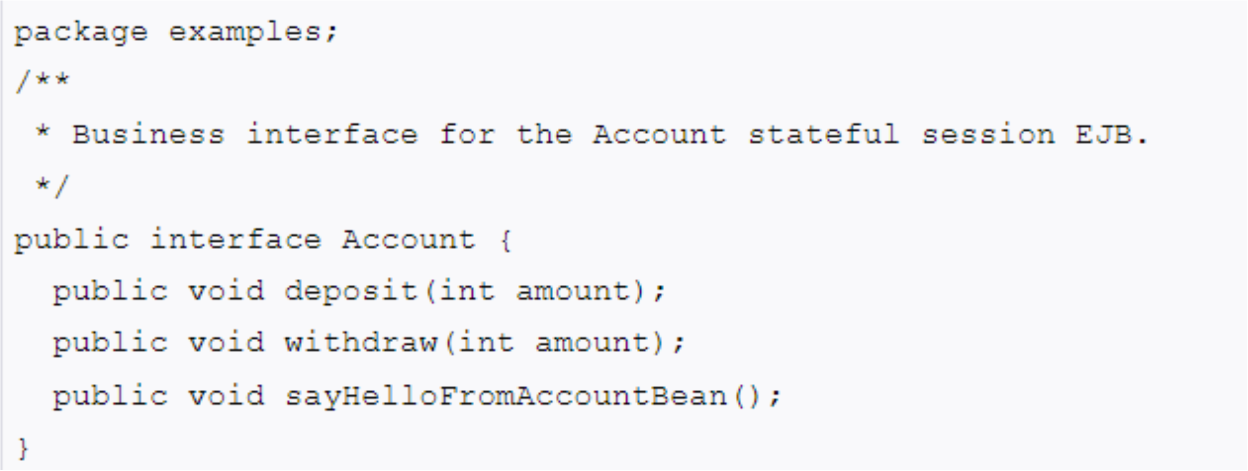
EJB application is deployed on the server, so it is called server-side component also.

EJB is like COM (Component Object Model) provided by Microsoft. But, it is different from Java Bean, RMI and Web Services.

CODE :

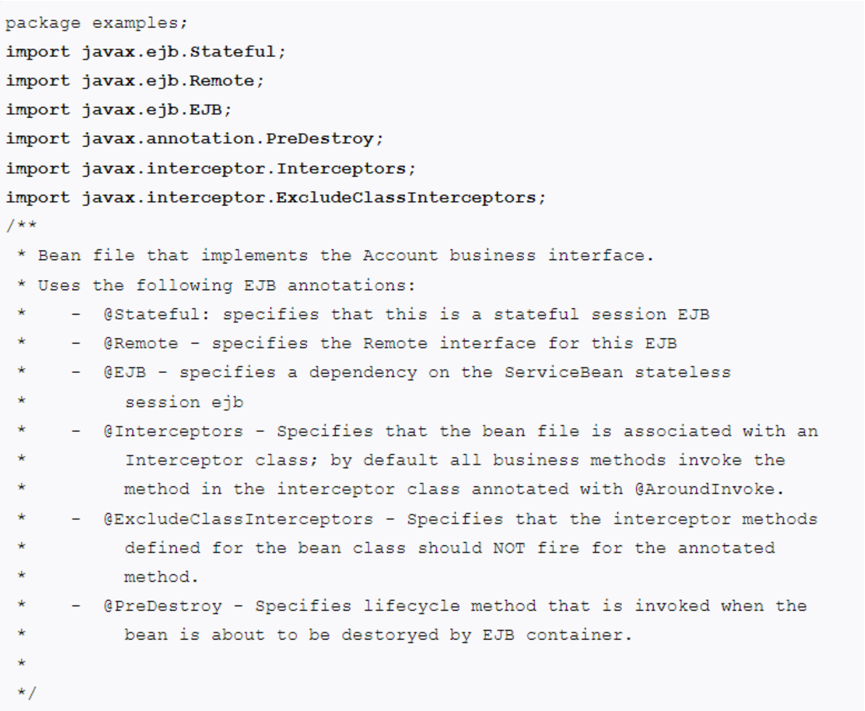
Example of a simple stateful EJB

- **Business Interface for AccountBean stateful session :**

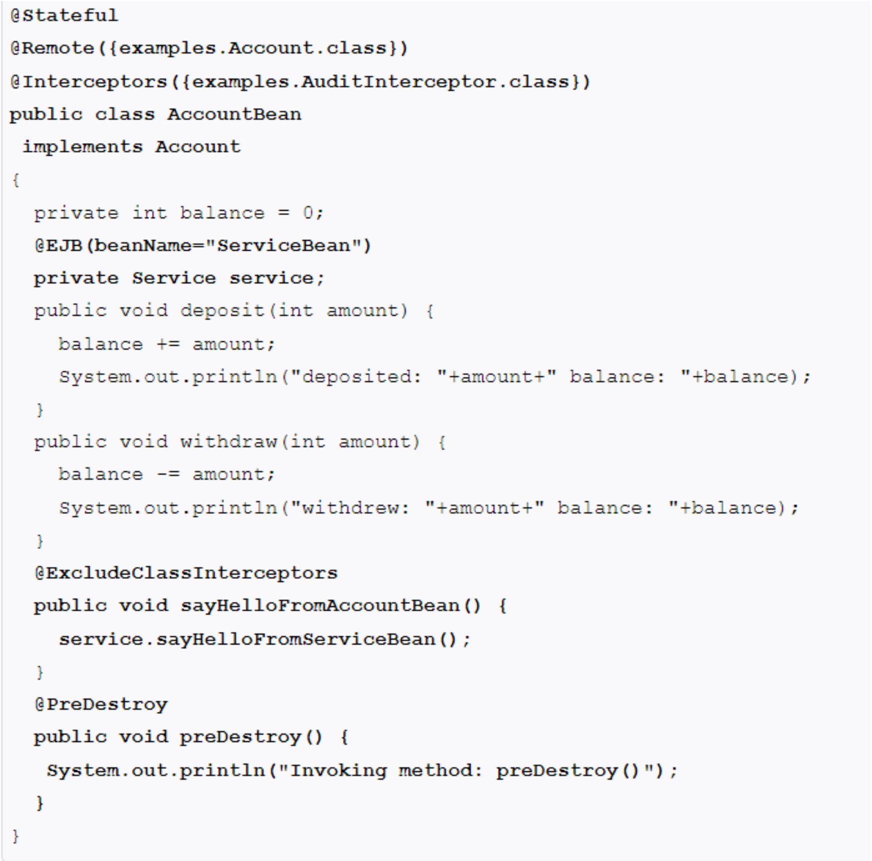


```
package examples;
/**
 * Business interface for the Account stateful session EJB.
 */
public interface Account {
  public void deposit(int amount);
  public void withdraw(int amount);
  public void sayHelloFromAccountBean();
}
```

- **Bean file that implements the Account interface :**



```
package examples;
import javax.ejb.Stateful;
import javax.ejb.Remote;
import javax.ejb.EJB;
import javax.annotation.PreDestroy;
import javax.interceptor.Interceptors;
import javax.interceptor.ExcludeClassInterceptors;
/**
 * Bean file that implements the Account business interface.
 * Uses the following EJB annotations:
 *    -  @Stateful: specifies that this is a stateful session EJB
 *    -  @Remote - specifies the Remote interface for this EJB
 *    -  @EJB - specifies a dependency on the ServiceBean stateless
 *         session ejb
 *    -  @Interceptors - Specifies that the bean file is associated with an
 *         Interceptor class; by default all business methods invoke the
 *         method in the interceptor class annotated with @AroundInvoke.
 *    -  @ExcludeClassInterceptors - Specifies that the interceptor methods
 *         defined for the bean class should NOT fire for the annotated
 *         method.
 *    -  @PreDestroy - Specifies lifecycle method that is invoked when the
 *         bean is about to be destoryed by EJB container.
 *
 */
```



```
@Stateful
@Remote({examples.Account.class})
@Interceptors({examples.AuditInterceptor.class})
public class AccountBean
 implements Account
{
  private int balance = 0;
  @EJB(beanName="ServiceBean")
  private Service service;
  public void deposit(int amount) {
    balance += amount;
    System.out.println("deposited: "+amount+" balance: "+balance);
  }
  public void withdraw(int amount) {
    balance -= amount;
    System.out.println("withdrew: "+amount+" balance: "+balance);
  }
  @ExcludeClassInterceptors
  public void sayHelloFromAccountBean() {
    service.sayHelloFromServiceBean();
  }
  @PreDestroy
  public void preDestroy() {
   System.out.println("Invoking method: preDestroy()");
  }
}
```

Result : Demonstrated the application of a simple stateful EJB.