Multiple ORM frameworks are there and each has their exclusive db methods

|  |  |  |  |
| --- | --- | --- | --- |
| **Hibernate** | **TopLink** | **MyBatis** | **EclipseLink** |
| hibernateSave(){...) | topLinkSave(){...) | myBatisSave(){...) | eclipseSave(){...) |
| HibernateUpdate(){..} | topLinkUpdate(){..} | myBatisUpdate(){..} | eclipseUpdate(){..} |
|  |  |  |  |

**In our application we will make our vendor specific DAO class and invoke the vendor specific ORM method.**

|  |  |  |  |
| --- | --- | --- | --- |
| **class HibernateDAO** | **Class DAOTopLink** | **Class DAOMyBatis** | **Class DAOEclipseLink** |
| HibernateDAO. hibernateSave() | DAOTopLink .topLinkSave() | DAOMyBatis. myBatisSave() | DAOEclipseLink .eclipseSave() |
|  |  |  |  |
| HibernateDAO. hibernateUpdate () | DAOTopLink .topLinkUpdate () | DAOMyBatis. myBatisUpdatee() | DAOEclipseLink .eclipseUpdate () |

**JPA is an interface which specifies the common rules which all the ORM providers should follow**

**All the above 4 ORM are implementing the JPA interface, so JPA has given standard method names for the DB methods as follows**

|  |
| --- |
| **Interface JPA{** |
| **Save();** |
| **Update();** |
| **Delete; }** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Hibernate implements JPA** | **TopLink implements JPA** | **MyBatis implements JPA** | **eclipseLink implements JPA** |
| Save(){...) | Save(){...) | Save(){...) | Save(){...) |
| Update(){..} | Update(){..} | Update(){..} | Update(){..} |
|  |  |  |  |
|  |  |  |  |

**For if i have hibernate as my vendor i will use**

**JPA jpadao=new Hibernate()**

**jpadao.save();**

**jpadao.update();**

**So next time if i switch to provider Toplink/MyBatis/.. then i don’t have to make any breaking changes in my DAO class , just simple plug and play and the same functions Jpa.save(), Jpa.update() will work for me.**

**JPA jpadao=new Toplink()**

**jpadao.save();**

**jpadao.update();**