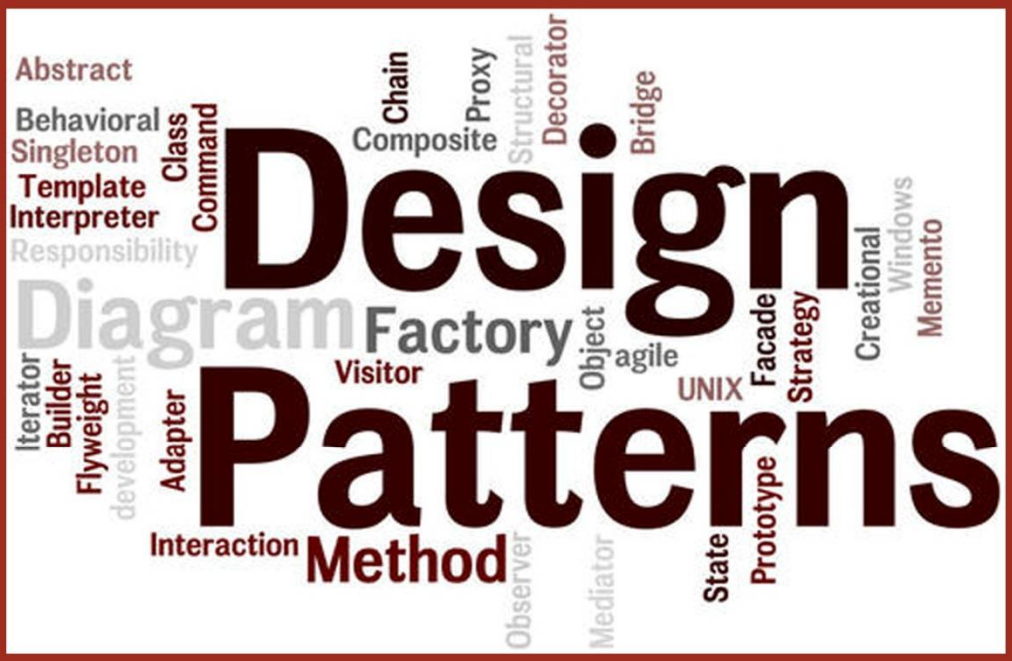
JAVA means DURGA SOFT

Design Patterns

Enterprise Application Level Design Patterns

2. DAO Factory Design Pattern



India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

DAO Factory Design Pattern

The DAO Factory design pattern is nothing but an extension to DAO Design Pattern. Generally we can use DAO Design Pattern in order to separate persistence logic from other logics of the application. Here implemented DAO class/ component contains persistence logic with respective to a single a database. If you want to work with multiple databases then go through the DAO Factory design pattern.

Def: DAO Factory design pattern is a class or component, which contains logic of constructing and returning one database software specific DAO class object based on the database software name we are provided i.e. DAO Factory design pattern is a combination of both Factory Pattern and DAO Design Pattern.

This design pattern helps the programmer to change the database software of the project with minimum changes done in JDBC persistence logic.

Ex:

//BComp1BaseDAO.java

```
public abstract class BComp1BaseDAO
{
	public abstract Connection getConnection();
	public abstract void closeConnection();
	public abstract int insert(-,-,-);
	public abstract List fetch(-,-);
	public static BComp1BaseDAO chooseDAO(String dbname)
	{
		if(dbname.equals("oracle"))
		{
			return new Comp1OracleDAO();
		}
		else if(dbname.equals("MsAccess"))
		{
			return new Comp1MsAccessDAO();
		}
		else
		{
			throw new Exception("Please Provide Existed D/B Name");
			return null;
		}
	}
};

public class  Comp1OracleDAO extends BComp1BaseDAO
{
```

```
        public Connection getConnection()
        {
                ----
                ----
        }
        public int insert(-,-,-)
        {
                ----
                ----
        }
        public List fetch(-,-)
        {
                ----
                ----
        }
        public void closeConnection()
        {
                ----
                ----
        }
}
public class  Comp1MsAccessDAO extends BComp1BaseDAO
{
        public Connection getConnection()
        {
                ----
```

```
                ----
        }
        public int insert(-,-,-)
        {
                ----
                ----
        }
        public List fetch(-,-)
        {
                ----
                ----
        }
        public void closeConnection()
        {
                ----
                ----
        }
}
public class BComp1
{
        BComp1BaseDAO dao=null;
        public BComp1()
        {
                dao=BComp1BaseDAO.chooseDAO("oracle");
```

```
        }
        public void bm1()
        {
                dao.getConnection();
                ---- // write some business logic
                ----
                // implement some persistence logic
                dao.insert(-,-,-);
                dao.closeConnection();
        }
}
public class BComp2
{
        BComp1BaseDAO dao=null;
        public BComp1()
        {
                dao=BComp1BaseDAO.chooseDAO("MsAccess");
        }
        public void bm2()
        {
                dao.getConnection();
                ---- // write some business logic
                ----
                // implement some persistence logic
                dao.insert(-,-,-);
                dao.closeConnection();
```

```
        }
}
```