

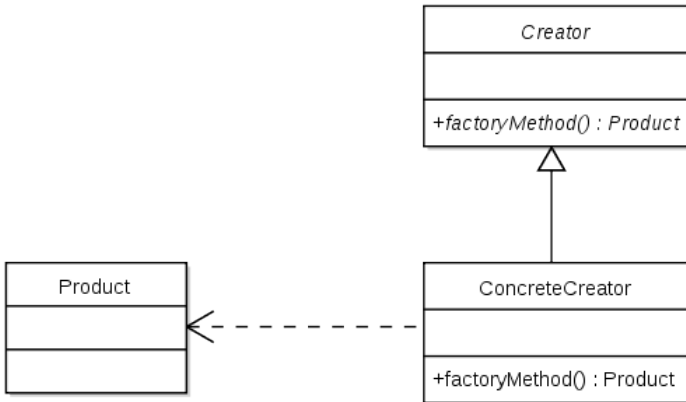
???????:

????????????, ??????????????. ?????????????????????.

Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.

?????????????:

1. ???????
  2. ?????????????????
- ????????????????, ????????:



????????????, ????????. Creator??????, ??????????????????. ??????????:

```

public interface IDbConnectionFactory {
    DbConnection CreateDbConnection(string connectionString);
}

public class MsSqlDbConnectionFactory : IDbConnectionFactory {
    public DbConnection CreateDbConnection(string connectionString)
    {
        DbConnection dbConnection = new SqlConnection();
        dbConnection.ConnectionString = connectionString;

        return dbConnection;
    }
}
    
```

????????????, ??IDbConnectionFactory????CreateConnection????, ?????????????????????, ??????????????. ?????????????????????????????????????DbConnection????????, ????.

????????UML?, factoryMethod????ConcreteCreator???. ???????, ?????????????????????, ?????????, ??????????????????????. MsSqlDbConnectionFactory??IDbConnectionFactory?CreateDbConnection?????????. ?????????????????????.

????????????????????, ??????????????. ???????, ???????OLEDB??????, ?????????????OLEDB????????, ?????????:

```

public class OleDbConnectionFactory : IDbConnectionFactory {
    public DbConnection CreateConnection(string connectionString) {
        DbConnection dbConnection = new OleDbConnection();
        dbConnection.ConnectionString = connectionString;

        return dbConnection;
    }
}
    
```

??????????, ?????????, ???????MS SQL???, ?????OLEDB???, ??????. ?????????, ??????????????  
 ?Client?????, Client????????????????????, ???????, ??????. ?????(?)????????, ??????. ??????????:

```
protected void Page_Load(object sender, EventArgs e)
{
    string connectionString = _connectionString;
    IDbConnectionFactory dbConnectionFactory = new MsSqlDbConnectionFactory();
    DbConnection dbConnection = dbConnectionFactory.CreateDbConnection(connectionString);
}
```

??Page\_Load????????????????????, Client??Page\_Load????CreateDbConnection?????????. ??????????????????????  
 ?.

**????: ADO.NET????(ADO.NET?????????)**

?ADO.NET?, ???Connection?????database?????, ?????????, ??????:

**▲ Inheritance Hierarchy**

- System.Object
- System.MarshalByRefObject
- System.ComponentModel.Component
- System.Data.Common.DbConnection**
- System.Data.EntityClient.EntityConnection
- System.Data.Odbc.OdbcConnection
- System.Data.OleDb.OleDbConnection
- System.Data.OracleClient.OracleConnection
- System.Data.SqlClient.SqlConnection

**Namespace:** System.Data.Common  
**Assembly:** System.Data (in System.Data.dll)

**▲ Syntax**

C# C++ F# VB

```
public abstract class DbConnection : Component,
    IDbConnection, IDisposable
```

????????????DbConnection, ??????. ?????????????????, ??????????????: IDbConnection. ?????????????????????  
 ????. IDbConnection????????????????????, DbConnection?????????????, ?????????????????????, ?????ADO.NET????  
 ??DbConnection???, ??????????. ???????, DbConnection????????????????, ??????????????????. ?????SqlConnection  
 ??????.

## ▲ Inheritance Hierarchy

System.Object  
 System.MarshalByRefObject  
 System.ComponentModel.Component  
 System.Data.Common.DbConnection  
**System.Data.SqlClient.SqlConnection**

**Namespace:** System.Data.SqlClient  
**Assembly:** System.Data (in System.Data.dll)

## ▲ Syntax

```

C# C++ F# VB
public sealed class SqlConnection : DbConnection,
    ICloneable
    
```

SqlConnection??DbConnection????, ??????MSSQL?????.????SqlConnection???????:

```

private void CompleteOpen();
private string ConnectionString_Get();
private void ConnectionString_Set(string value);
private void CopyFrom(SqlConnection connection);
public SqlCommand CreateCommand();
protected override DbCommand CreateDbCommand();
private static CodeAccessPermission CreateExecutePermission();
protected override void Dispose(bool disposing);
private void DisposeMe(bool disposing);
    
```

??Reflector??System.Data.dll????, ?????Command?????: CreateCommand?CreateDbCommand. ??  
 CreateCommand????????, ?????, ?CreateDbCommand?????????. ??????????????, CreateDbCommand?????  
 Command????, ???protected?, ??????. ???DbConnection????????CreateCommand????, ???DbConnection???  
 ??CreateCommand??CreateDbCommand????, ?????????DbCommand??.

```

public DbCommand CreateCommand()
{
    return this.CreateDbCommand();
}
    
```

?????????, CreateCommand????Command?????, ??????????????CreateDbCommand?, ??????????????????  
 ??????, ?????????.

```
protected override DbCommand CreateDbCommand()
{
    DbCommand command = null;
    IntPtr ptr;
    Bid.ScopeEnter(out ptr, "<prov.DbConnectionHelper.CreateDbCommand|API> %d#\n", this.ObjectID);
    try
    {
        command = this.ConnectionFactory.ProviderFactory.CreateCommand();
        command.Connection = this;
    }
    finally
    {
        Bid.ScopeLeave(ref ptr);
    }
    return command;
}
```

???SqlConnection?????DbConnection?CreateDbCommand????????, ?????, ?????????????????????MSSQL????  
 ??????. ?????????????????, ??????:

1. ?????????????=> DbConnection?CreateDbCommand
2. ?????????????????=> SqlConnection?CreateDbCommand