## 2021/8/10

**笔记本:** diary
**创建时间:** 2021/8/10 17:06 **更新时间:** 2021/8/10 17:21
**作者:** 134exetj717
**URL:** https://zhidao.baidu.com/question/258446552.html

---

- 开发之前，我突然傻傻分不清UI和原型设计的区别：可怕~
  - 原型设计是基于UI和UX之上的，原型设计是视觉和逻辑的总和；UI是视觉效果，用户界面的设计；UX是内部逻辑，代表不同界面之间的关系以及产品的功能。
  - 举个例子：当你带着自己的项目去谈生意时，你需要告诉客户，项目是怎么样的？（即原型设计）那么首先，你会让UI来展示项目的界面；然后，你会让UX来讲解你的项目功能逻辑。

---

三层架构分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）再加上实体类库（Model）

```
DAL : Data access layer 数据访问层
BLL : Business Logic Layer 业务逻辑层
UI : User Interface layer 表示层
```

1、实体类库（Model），主要存放数据库中的表字段。

操作：

（1）先建立实体类库Model，打开项目，在解决方案中右键--》添加--》新建项目--》选中类库--》改名Model--》确定

（2）选中Model类库--》Shift+ALT+C--》建立实体类。UserInfo类

```
namespace Model
    {
       public  class UserInfo
        {
            public string  UserName { get; set; }
            public string  Password { get; set; }
        }
    }
```

2、数据访问层（DAL），主要是存放对数据类的访问，即对数据库的添加、删除、修改、更新等基本操作

操作：

（1）先建立数据访问层类库DAL，打开项目，在解决方案中右键--》添加--》新建项目--》选中类库--》改名DAL--》确定

（2）在DAL中添加对Model的引用，选中DAL--》Alt+P+R--》解决方案--》项目--》选中MOdel--》确定

（3）在DAL中添加对system.configuration的引用，选中DAL--》Alt+P+R--》程序集--》框架--》选中System.configuration--》确定

（4）建立数据访问类，选中DAL--》Shift+ALT+C--》建立数据访问类。UserDB类

```
using System.Configuration;
    using Model;
    using System.Data;
    using System.Data.SqlClient;

    namespace DAL
    {
        class UserDB
        {
            private string connString =
ConfigurationManager.ConnectionStrings["connString"].ToString();
            public int AddUser(UserInfo userInfo)
            {
                //对数据库进添加一个用户操作
                string commandText = "insert into UserInfo
(userName,Password)values(@userName,@Password)";
                SqlParameter[] paras = new SqlParameter[]
                {
               new SqlParameter ("@userName",userInfo.UserName ),
               new SqlParameter ("@Password",userInfo.Password )
                };
                return SqlHelper.ExecuteNonQuery(connString, CommandType.Text,
commandText, paras);
            }
        }


    //添加其他对数据库操作
    }
```

3、业务逻辑层（BLL）对传送数据进行逻辑判断分折，并进行传送正确的值。

（1）先建立业务逻辑层类库BLL，打开项目，在解决方案中右键--》添加--》新建项目--》选中类库--》改名BLL--》确定

（2）在BLL中添加对Model、DAL的引用，选中BLL--》Alt+P+R--》解决方案--》项目--》选中MOdel、DAL--》确定

（3）建立业务逻辑类，选中BLL--》Shift+ALT+C--》建立业务逻辑类。LoginManager类

```
using DAL;
    using Model;

    namespace BLL
    {
        public class LoginManager
        {
            private UserDB userDB = new UserDB();
            public bool Add(UserInfo userInfo, out string messageStr)
            {
```

```
                messageStr = "";//返回界面层添加用户返回信息
                bool isSuccess = false;
                if (userInfo.UserName.Trim().Length != 0)//判断从传递来的username是否为空
                {
                    if (userDB.IsEquals(userInfo))//传给DALl操作判断数据库中是否有重复值
                    {
                        userDB.AddUser(userInfo);//传给DAL操作增加一个新用户
                        isSuccess = true;
                    }
                    else
                        messageStr = "有相同的值";
                }
                else
                {
                    messageStr = "不能为空";

                }
                return isSuccess;//返回界面层是否添加成功
            }
        }
    }
```

5、表现层（UI）即用户界面层

（1）在UI中添加对Model、BLL的引用，选中UI--》Alt+P+R--》解决方案--》项目--》选中MOdel、BLL--》确定

（2）编写代码传递数据给BLL层。

```
UserInfo userInfo;
            LoginManager lm = new LoginManager();
            private void btnAdd_Click(object sender, EventArgs e)
            {
                userInfo = new UserInfo()
                {
                    UserName = txtUserName.Text.Trim(),
                    Password = txtPassword.Text.Trim()
                };
                string messageStr = "";

                if (lm.Add(userInfo, out  messageStr))
                {
                    MessageBox.Show("添加成功");
                }
                else
                {
                    MessageBox.Show(messageStr);
                    txtUserName.Focus();
                }

            }
        }
```