

## 作业五 业务逻辑层设计

业务逻辑层类库 BLL 由 4 个核心类组成，分别为 RoleManager.cs、UserManager.cs、RoomManager.cs 和 ClientManager.cs。

### 1.RoleManager 类

RoleManager 类，用于保存当前登录用户的相关信息。

```
public class RoleManager
{
    public static User curUser = new User();
}
```

【分析】用户登录成功后，把从 user（用户信息表）中查询到数据（User 类型对象）存储到 RoleManager 类的 curUser 字段中，用于在入住登记等界面中获取当前用户（即操作员）。

### 2.UserManager 类

UserManager 类用于对用户进行管理，该类封装了 2 个方法，分别为 GetUser()方法和 AddUser()方法。

（1）GetUser()方法。用于根据用户名和密码获取用户，代码如下：

```
public static User GetUser(string name, string pwd)
{
    User user = null;
    string sql = "select * from user where userName=\'" + name + "\' and userPwd=\'" +
pwd + "\'";
    MySqlDataReader reader= DBOper.GetReader(sql);
    if (reader.Read ())
    {
        user = new User();
        user.Name =Convert.ToString ( reader["userName"]);
        user.Pwd = Convert.ToString(reader ["userPwd"]);
        user.Role = Convert.ToInt32(reader ["role"]);
    }
    reader.Close();
    return user;
}
```

【分析】用户在登录时，输入用户名和密码，用户名和密码作为 GetUser()方法参数，然后从数据库 user（用户信息表）中查询该用户，如果存在，则返回 User 类型对象。

（2）AddUser()方法。用于将新用户信息添加到数据库中的 user 表中，代码如下：

```
public static bool AddUser(User user)
{
    string sql = "insert into user values(@name,@pwd,@role)";
    MySqlParameter p1 = new MySqlParameter("@name", user.Name );
    MySqlParameter p2 = new MySqlParameter("@pwd", user.Pwd);
    MySqlParameter p3 = new MySqlParameter("@role", user.Role);
    if (DBOper.ExecuteCommand(sql, p1, p2, p3)==1)
    {
        return true;
    }
}
```

```

        else
        {
            return false;
        }
    }
}

```

【分析】在管理员添加新用户时，输入的用户名、密码和权限数据存储在 User 类型对象属性中，并作为 AddUser()方法的参数传入，如果添加成功，则返回 true，否则返回 false。

(3) GetUserInfo()方法。用于从 user 表中获取用户信息，返回值为 DataTable，代码如下：

```

public static DataTable GetUserInfo()
{
    string sql = @"select userName as 用户名,userPwd as 用户密码,role as 用户权限 from
user";

    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}

```

【分析】在管理员修改/删除用户时，在界面加载时将所有用户信息获取并返回到数据表中。

(4) UpdateUser()方法。用于保存修改后的用户信息。

```

public static bool UpdateUser(User user)
{
    string sqlUpdateUser = @"update user set userPwd=@userPwd,role=@role where
userName=@userName";

    MySqlParameter p1 = new MySqlParameter("@userPwd", user.Pwd );
    MySqlParameter p2 = new MySqlParameter("@role", user.Role );
    MySqlParameter p3 = new MySqlParameter("@userName", user.Name );
    MySqlParameter[] paramArray = new MySqlParameter[] { p1, p2, p3 };
    if (DBOper.ExecuteCommand(sqlUpdateUser, paramArray) == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

【分析】在管理员修改/删除用户时，当点击修改按钮时，将修改后的数据封装成 User 类型的对象，并执行数据库修改操作，其中用户名为主键，不能修改。

(5) DeleteUser()方法。用于删除用户信息。

```

public static bool DeleteUser(User user)
{
    string sqlDeleteUser = @"delete from user where userName=@userName";
    MySqlParameter p1 = new MySqlParameter("@userName", user.Name);
    MySqlParameter[] paramArray = new MySqlParameter[] { p1 };
    if (DBOper.ExecuteCommand(sqlDeleteUser, paramArray) == 1)
    {

```

```

        return true;
    }
    else
    {
        return false;
    }
}

```

【分析】在管理员修改/删除用户时，当点击“删除”按钮时，将查询到该用户并删除。

### 3. RoomManager 类

RoomManager 类用于对房间进行管理。该类中封装了 7 个方法，分别为 GetRoomInfo() 方法、InsertRoomInfo() 方法、GetUseRoomInfo() 方法、GetInId() 方法、InsertCheckOutRoomInfo() 方法、GetRoomInfo() 方法和 GetRoomUseInfo() 方法。

(1) GetRoomInfo() 方法。用于从 room 表中获取空房信息，返回值为 DataTable 类型，代码如下：

```

public static DataTable GetRoomInfo()
{
    string sql = @"select roomId as 房间号,roomType as 房间类型,roomFloor as 层数, Price
as 价格,personNum as 可入住人数,inPerson as 已入住人数,note as 备注 from room where
inPerson=0";
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}

```

【分析】在入住登记时，入住登记界面中将显示全部房间信息，即从 room（房间信息表）中查询入住人数 inPerson 值为 0 的数据，并将查询到的数据表返回。

(2) InsertRoomInfo() 方法。用于保存入住信息到 checkin 表中，并且更新 room 表中的入住人数 inPerson 值，返回值为 bool 类型，代码如下：

```

public static bool InsertRoomInfo(CheckInRoom registerRoomInfo)
{
    string sqlInsert = @"insert into checkin values(null,@roomId,@price,@foregift,
@inTime,@outTime,@clientName,@sex,@phone,@certType,@certId,@address,
@personNum,@Oper,@delMark)";
    MySqlParameter p1 = new MySqlParameter("@roomId", registerRoomInfo.RoomId);
    MySqlParameter p2 = new MySqlParameter("@price", registerRoomInfo.Price);
    MySqlParameter p3 = new MySqlParameter("@foregift", registerRoomInfo.Foregift);
    MySqlParameter p4 = new MySqlParameter("@inTime", registerRoomInfo.InTime);
    MySqlParameter p5 = new MySqlParameter("@outTime", registerRoomInfo.OutTime);
    MySqlParameter p6 = new MySqlParameter("@clientName", registerRoomInfo.ClientName);
    MySqlParameter p7 = new MySqlParameter("@sex", registerRoomInfo.Sex);
    MySqlParameter p8 = new MySqlParameter("@phone", registerRoomInfo.Phone);
    MySqlParameter p9 = new MySqlParameter("@certType", registerRoomInfo.CertType);
    MySqlParameter p10 = new MySqlParameter("@certId", registerRoomInfo.CertId);
    MySqlParameter p11 = new MySqlParameter("@address", registerRoomInfo.Address);
    MySqlParameter p12 = new MySqlParameter("@personNum", registerRoomInfo.PersonNum);
    MySqlParameter p13 = new MySqlParameter("@Oper", registerRoomInfo.Oper);
}

```

```

        MySqlParameter p14 = new MySqlParameter("@delMark", registerRoomInfo.DelMark);
        MySqlParameter[] paramArray = new MySqlParameter[] { p1, p2, p3, p4, p5, p6, p7, p8, p9,
        p10, p11, p12, p13, p14 };
        string sqlUpdate = "update room set inPerson=@inPerson where roomId=@roomId";
        MySqlParameter n1 = new MySqlParameter("@roomId", registerRoomInfo.RoomId);
        MySqlParameter n2 = new MySqlParameter("@inPerson", registerRoomInfo.PersonNum);
        if (DBOper.ExecuteCommand(sqlInsert, paramArray) == 1 && DBOper.ExecuteCommand(sqlUpdate,
        n1, n2) == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

【分析】在入住登记时，操作员在入住登记界面中输入住客的相关信息，并且将信息添加到 checkin（入住信息表）中，并且将入住人数保存到 room（房间信息表）中的入住人数 inPerson 字段中，即表明该房间已不是空房间。如果以上两个表的操作都成功，则返回 true，否则返回 false。

（3）GetUseRoomInfo()方法。用于从 checkin 表中获取入住信息，其中 delMark=0 表示已入住并且没有退房，返回值为 DataTable 类型，代码如下：

```

public static DataTable GetUseRoomInfo()
{
    string sql = @"select roomId as 房间号,price as 价格,foregift as 押金, inTime as 入住时间,outTime as 退房时间,clientName as 住客姓名, sex as 性别,phone as 电话号码,certType as 证件类型, certId as 证件号码, address as 地址, personNum as 入住人数, Oper as 登记者 from checkin where delMark=0";
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}

```

【分析】在退房登记时，在退房登记界面初始化时，需要获取所有已入住并且没有退房的入住信息，即从 checkin（入住信息表）中查询 delMark 值为 0 的全部数据并返回。

（4）GetInId()方法。用于从 checkin 表中根据房间号 roomId 查找 delMark=0 的退房房间的 inId，返回值为 DataTable 类型，代码如下：

```

public static DataTable GetInID(string roomId)
{
    string sql = @"select inId from checkin where delMark=0 and roomId=" + roomId;
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}

```

【分析】在退房登记时，当操作员点击“保存”按钮时，需要将退房信息保存，其中 inId 的值需要在 checkin（入住信息表）中根据房间号 roomId（且 delMark 为 0，即当前已入住并且没有退房）字段查询并返回。

(5) InsertCheckOutRoomInfo()方法。用于保存退房信息到 checkout 表中，更新 checkin 表中的 delMark 值为 1，并且更新 room 表中的 InPerson 为 0，返回值为 bool 类型，代码如下：

```
public static bool InsertCheckOutRoomInfo(CheckOutRoom checkOutRoomInfo)
{
    string sqlInsert = @"insert into checkout
values(@outId,@inId,@outTime,@roomId,@clientName,@inTime,@price,@foregift,@total,@account,@note,@oper)";

    string sqlUpdateRegister = "update checkin set delMark=1 where inId=@inId";
    string sqlUpdateRoom = "update room set inPerson=0 where roomId=@roomId";
    MySqlParameter p1 = new MySqlParameter("@outId", checkOutRoomInfo.OutId);
    MySqlParameter p2 = new MySqlParameter("@inId", checkOutRoomInfo.InId);
    MySqlParameter p3 = new MySqlParameter("@roomId", checkOutRoomInfo.RoomId);
    MySqlParameter p4 = new MySqlParameter("@price", checkOutRoomInfo.Price.ToString());
    MySqlParameter p5 = new MySqlParameter("@foregift",
checkOutRoomInfo.Foregift.ToString());
    MySqlParameter p6 = new MySqlParameter("@total", checkOutRoomInfo.Total.ToString());
    MySqlParameter p7 = new MySqlParameter("@account",
checkOutRoomInfo.Account.ToString());
    MySqlParameter p8 = new MySqlParameter("@inTime", checkOutRoomInfo.InTime);
    MySqlParameter p9 = new MySqlParameter("@outTime", checkOutRoomInfo.OutTime);
    MySqlParameter p10 = new MySqlParameter("@clientName", checkOutRoomInfo.ClientName);
    MySqlParameter p11 = new MySqlParameter("@oper", checkOutRoomInfo.Oper);
    MySqlParameter p12 = new MySqlParameter("@note", checkOutRoomInfo.Note);
    MySqlParameter[] paramArray = new MySqlParameter[] { p1, p2, p3, p4, p5, p6, p7, p8, p9,
p10, p11, p12 };
    MySqlParameter n1 = new MySqlParameter("@inId", checkOutRoomInfo.InId);
    MySqlParameter n2 = new MySqlParameter("@roomId", checkOutRoomInfo.RoomId);
    if (DBOper.ExecuteCommand(sqlInsert, paramArray) == 1 &&
        DBOper.ExecuteCommand(sqlUpdateRegister, n1) == 1 &&
        DBOper.ExecuteCommand(sqlUpdateRoom, n2) == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

【分析】在退房登记时，当操作员点击“保存”按钮时，将所有信息保存到 checkout（退房信息表）中，并且修改 checkin（入住信息表）中的 delMark 值为 1，即表示已退房，同时，修改 room（房间信息表）中的入住人数 inPerson 值为 0，即表示该房间为房间。如果对以上三个表的操作都成功，则返回 true，否则返回 false。

(6) GetRoomInfo()方法。用于从 room 表中根据 roomId 查询房间信息，返回值为 DataTable 类型，代码如下：

```
public static DataTable GetRoomInfo(string roomId)
{
    string sql = @"select * from room where roomId=" + roomId;
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}
```

【分析】在住客信息查询界面中，操作员如果要修改查询到的住客信息的入住人数时，需要验证修改后的值不能大于该房间允许入住的人数，因此，需要在 room（房间信息表）中根据 roomId 查询出该房间信息并返回。

（7）GetRoomUseInfo()方法。用于从 room 表中查询非空房间信息，即 inPeron 值不为 0 的房间信息，返回值为 DataTable 类型，代码如下：

```
public static DataTable GetRoomUseInfo()
{
    string sql = @"select roomId as 房间号,roomType as 房间类型,roomFloor as 层数, Price
as 价格,personNum as 可入住人数,inPerson as 已入住人数, note as 备注 from room where
inPerson!=0";
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}
```

【分析】在客户信息查询界面加载中，需要从 room（房间信息表）查询已入住（即入住人数 inPerson 值为 0）的房间信息。

#### 4.ClientManager 类

ClientManager 类用于对住客信息进行管理。该类封装 4 个方法，分别为 GetClientInfo()方法、GetAllClient()方法、UpdateRegister()方法和 GetClient()方法。

（1）GetClientInfo()方法。用于从 checkin 表中获取住客信息，返回值为 DataTable 类型，代码如下：

```
public static DataTable GetClientInfo()
{
    string sql = @"select roomId as 房间号,price as 价格,foregift as 押金, inTime as 入住时间,outTime as 退房时间,clientName as 住客姓名,sex as 性别,phone as 电话号码,certType as 证件类型, certId as 证件号码,address as 地址,personNum as 入住人数, Oper as 登记者,delMark as 是否退房,inId from checkin";
    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}
```

【分析】在住客信息查询界面加载时，需要从 checkin（入住信息表）中查询所有数据。

（2）GetAllClient()方法。用于从 checkin 表中根据姓名模糊查询住客信息，返回值为 DataTable 类型，代码如下：

```
public static DataTable GetALLClient(string clientName)
{
    string sql = @"select roomId as 房间号,price as 价格,foregift as 押金, inTime as 入住时间,outTime as 退房时间,clientName as 住客姓名,sex as 性别,phone as 电话号码,certType as 证件类型, certId as 证件号码,address as 地址,personNum as 入住人数, Oper as 登记者,delMark as 是否退房,inId from checkin where clientName like '%" + clientName + "%'";
```

```

        DataTable dt = DBOper.GetDataTable(sql);
        return dt;
    }

```

【分析】在住客信息查询界面中，可以根据操作员输入的住客姓名进行模糊查询。

(3) UpdateRegister()方法。用于在 checkin 表中保存住客信息，返回值为 bool 类型，代码如下：

```

public static bool UpdateRegister(CheckInRoom registerRoomInfo)
{
    string sqlUpdateRegister = @"update checkin set
clientName=@clientName,sex=@sex,phone=@phone,certType=@certType,
certId=@certId,address=@address,personNum=@personNum where inId=@inId";

    MySqlParameter p1 = new MySqlParameter("@clientName", registerRoomInfo.ClientName);
    MySqlParameter p2 = new MySqlParameter("@sex", registerRoomInfo.Sex);
    MySqlParameter p3 = new MySqlParameter("@phone", registerRoomInfo.Phone);
    MySqlParameter p4 = new MySqlParameter("@certType", registerRoomInfo.CertType);
    MySqlParameter p5 = new MySqlParameter("@certId", registerRoomInfo.CertId);
    MySqlParameter p6 = new MySqlParameter("@address", registerRoomInfo.Address);
    MySqlParameter p7 = new MySqlParameter("@personNum", registerRoomInfo.PersonNum);
    MySqlParameter p8 = new MySqlParameter("@inId", registerRoomInfo.InId);
    MySqlParameter[] paramArray = new MySqlParameter[] { p1, p2, p3, p4, p5, p6, p7, p8 };
    if (DBOper.ExecuteCommand(sqlUpdateRegister, paramArray) == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

【分析】在住客信息查询界面中，操作员可以将查询到的住客姓名等信息进行修改并保存在 checkin（入住信息表）中。

(4) GetClient()方法。用于从 checkin 表中根据 roomId 查找住客信息，返回值为 DataTable 类型，代码如下：

```

public static DataTable GetClient(string roomId)
{
    string sql = @"select roomId as 房间号,price as 价格,foregift as 押金, inTime as 入住时间,outTime as 退房时间,clientName as 住客姓名, sex as 性别,phone as 电话号码,certType as 证件类型, certId as 证件号码,address as 地址,personNum as 入住人数, Oper as 登记者 from checkin
where delMark=0 and roomId =" + roomId;

    DataTable dt = DBOper.GetDataTable(sql);
    return dt;
}

```

【分析】在客户信息查询界面中，需要从 checkin（入住信息表）中根据用户选择的房间号 roomId 查询出住客信息。