### 芯达通二期代码ReView

完成目标：

1. 逐步完成项目架构整理 (新增或修改代码进行按要求开发)

2、面向对象开发培训

### 一、代码结构整理及代码规范约束：

Needs.Wl.Models

|---Enums 基础枚举配置，命名空间：Needs.Wl.Models.Enums

|---Extends 对象的扩展函数，命名空间：Needs.Wl.Models

|---Interfaces 接口，命名空间：Needs.Wl.Models

|---Models 基础对象 ，命名空间：Needs.Wl.Models

|--Hanlders 委托类，与相关对象文件存放在一起 命名空间：Needs.Wl.Models.Hanlders

|--EventArgs 事件参数，与相关对象文件存放在一起 命名空间：Needs.Wl.Models.EventArgs

|---Views 基础View 命名空间：Needs.Wl.Models.Views

以财务模块为例

Needs.Wl.Finance.Services

|---Enums 枚举配置,命名空间：Needs.Wl.Finance.Services.Enums

|---Extends 对象的扩展函数 ,命名空间：Needs.Wl.Finance.Services

|---Interfaces 接口,命名空间：Needs.Wl.Finance.Services

|---Models 业务对象,命名空间：Needs.Wl.Finance.Services.Models

|--Hanlders 委托类,命名空间：Needs.Wl.Finance.Services.Hanlders

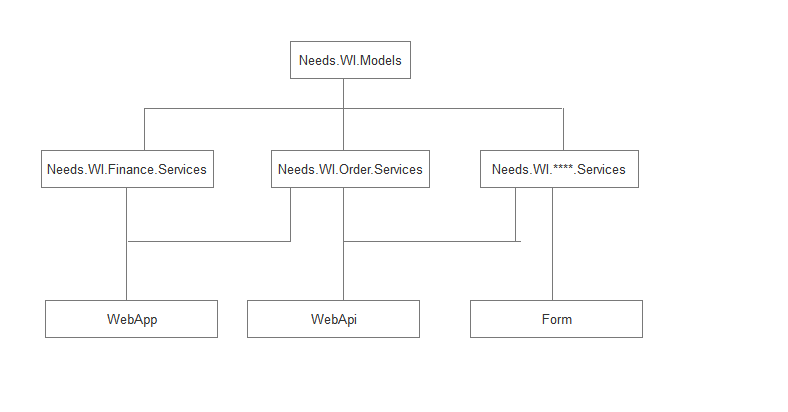
|--EventArgs 事件参数,命名空间：Needs.Wl.Finance.Services.EventArgs

|---Views,命名空间：Needs.Wl.Finance.Services.Views

|--PageModels/FormModels 页面视图类（如：列表中的字段、表单中的字段，类似MVC中的Models） ,命名空间：Needs.Wl.Finance.Services.PageModels

|---按业务逻辑划分的文件夹

### 二、继承、引用关系图



### 三、对象的拓展

1、继承

2、接口

3、拓展函数

### 四、纪律

1、严谨在Needs.Wl.Models层对象中随意添加属性、函数。

2、数据库添加字段、对象添加属性，必须通过会议，紧急时添加后，必须事后代码ReView。

### 文件创建相关总结

一、功能区文件

说明：按功能进行文件夹的创建，能体现出实现的功能。

举例：Client（客户信息的相关功能）； Classify（产品归类相关功能）

二、页面文件与命名

说明：子文件目录与代码文件目录要体现出树形结构，一般与对象的层次结构对应；代码文件的命名根据要实现的功能或要展示的数据命名。

举例：

Client

|--Address

|--List.aspx

|--Edit.aspx

|--Supplier

|--List.aspx

|--Edit.aspx

|--List.aspx

|--Edit.aspx

三、列表显示的规范

举例说明：1、列表页面钟，不固定或无法预知长度的文本，要左对齐与自动换行。

2、日期格式，如无特殊要求，格式为：yyyy-MM-dd HH:ss:mm

3、列表数据的排序

对象创建

一、对象

什么是对象，不再描述。

每个事物都要自己的质（状态）与量（数值），而且一定是变化的，如果一个事物是一直不变化的，那么这个对象将在系统中是配置或固定值的形式出现在系统中。

如，张三这个人，每天都在发生变化。如年龄、体重，健康状态等。

事物每时每刻都在发生变化（变更），变化体现在两种形式，一是量的变化（数值），一是质的变化（状态），比如，水，蒸发变为水蒸气，结冰变为冰，属于质的变更，比如，水温度，-1度，0度，100度，属于量（数值）的变更。

对象的数值与状态一定有初始状态（值）， 初始状态、值使用对象的构造函数进行初始化。

二、对象量（数值）的变更

对象的量的变更，具体是讲对象某一属性的量的变化，如水的温度，从0度变更为10度，如客户的跟单员、业务员的变更；对象的量的变更一定要有日志记录。

日志记录的规范描述包含但不限于：时间、人、事件（做了什么事情）、地点、原因、结果、下一步要做的事情等。

对象的量的变更的通知：事件。其他对象要想取得该对象的变更通知，一定要注册该事件。

二、对象质（状态）的变更

对象状态的变更有两种，一是通过外界的触发（或其他原因）产生变化，也就一定对应某一个功能；一是通过内在的原因产生变化。

外界触发：在系统中对应某一个功能，在界面中有一个触发、或在系统中通过其他对象的变更进行的触发（这也体现了万事万物都是相互关联的这个道理）。

内在原因：事物的自身的生命周期，如人的生老病死等，客户长期不进行付款等。

对象的状态变更一定要有日志记录， 日志记录的规范描述包含但不限于：时间、人、事件（做了什么事情）、地点、原因、结果、下一步要做的事情等。

对象的状态变更的通知：事件。其他对象要想取得该对象的变更通知，一定要注册该事件。

特别说明，不同类型（type）的事事物，如男女、海关汇率、实时汇率 ，也属于质的变更。

对象变更的代码参考如下：

public virtual void OnSealed()

{

if (this != null && this.Sealed != null)

{

this.Sealed(this, new SealedHanlderEventArgs(this.Order.ID));

}

}

public void Seal()

{

using (Layer.Data.Sqls.ScCustomsReponsitory reponsitory = new Layer.Data.Sqls.ScCustomsReponsitory())

{

//更新入库通知状态

reponsitory.Update<Layer.Data.Sqls.ScCustoms.EntryNotices>(new { EntryNoticeStatus = EntryNoticeStatus.Sealed }, item => item.ID == this.ID);

}

this.OnSealed();

}

三、对象变更的前与后

量的变更委托的命名： [对象][属性] ChangedHanlder,[对象][属性] ChangingHanlder

质的变更委托的命名： [对象][状态]edHanlder；[对象][状态]ingHanlder

量的变更事件的命名： [属性] Changed,[属性] Changing

质的变更事件的命名： [状态名称]ed；[状态名称]ing

代码注释规范：

delegate： 表示用于处理的方法 Order.Returned 事件

EventArgs：订单退回事件参数

event: 当【对象】【状态（如退回）】后、前发生。

三、对象命名与按状态、类型拓展新对象

以订单为例，订单有很多状态，Order,CancledOrder。CancledOrder继承Order, 订单有很多访问者，UserCancledOrder:CancledOrder,AdminCancledOrder:CancledOrder。

汇率（type 的如何封装 ），实时汇率、海关汇率。

以上为建模的参考。

四、对象的持久化

不同状态的对象的持久化，在对象中用状态属性表示，如订单状态:OrderStatus,可持久化在一个数据库的表中。对应的对象的设计与View视图的查询一定要掌握。

不同类的类型对象的持久化，在对象中用type（结构、值类型、枚举）属性表示，持久化在数据库的一个表中。对应的对象的设计与View视图的查询一定要掌握。

五、对象与页面操作

对象的变化，需要外界触发的，在系统中体现在页面上的一个功能按钮或其他对象的事件通知。

界面中输入的验证等不再描述。

六、界面设计

可以多看书、产品设计、看一些好的项目的设计，只能这样自我总结。