一、设计原则

1.模块定义

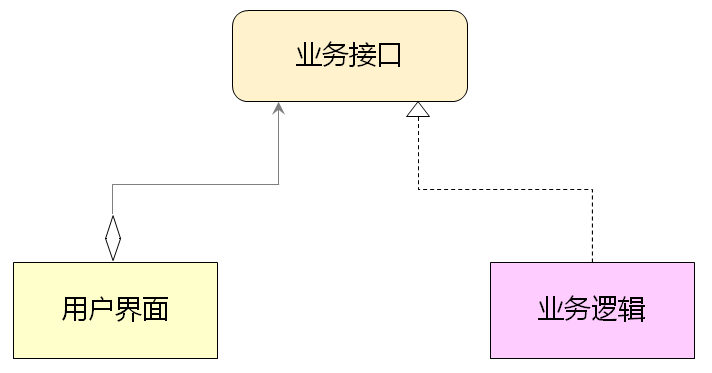
基本程序架构一般包括：

用户界面模块（UI）

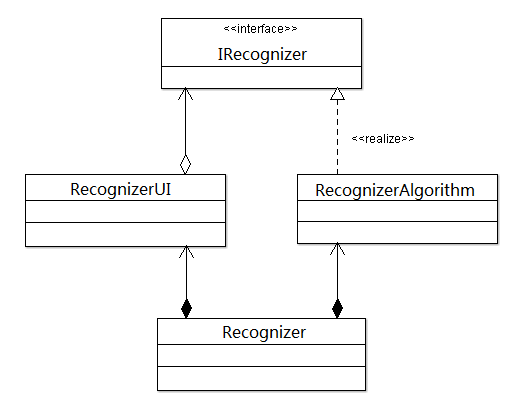
接收用户输入及呈现数据

业务逻辑模块（Business Logic）

根据用户需求处理数据，这里的业务逻辑就是识别算法及数据存储



二、数字识别应用程序的整体架构



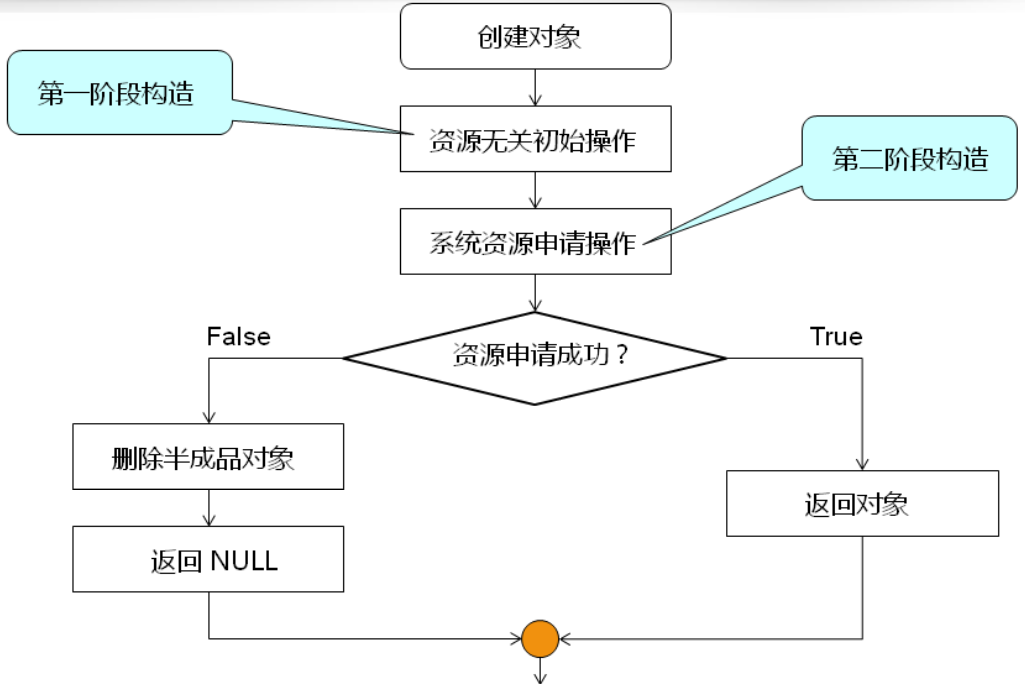
**三、数字识别中使用到的设计模式**

**1.二阶构造模式**

工程开发中的构造过程可分为：

1. 资源无关的初始化操作
   1. 不可能出现异常情况的操作
2. 需要使用系统资源的操作
   1. 可能出现异常情况，如：内存申请，访问文件

二阶构造：



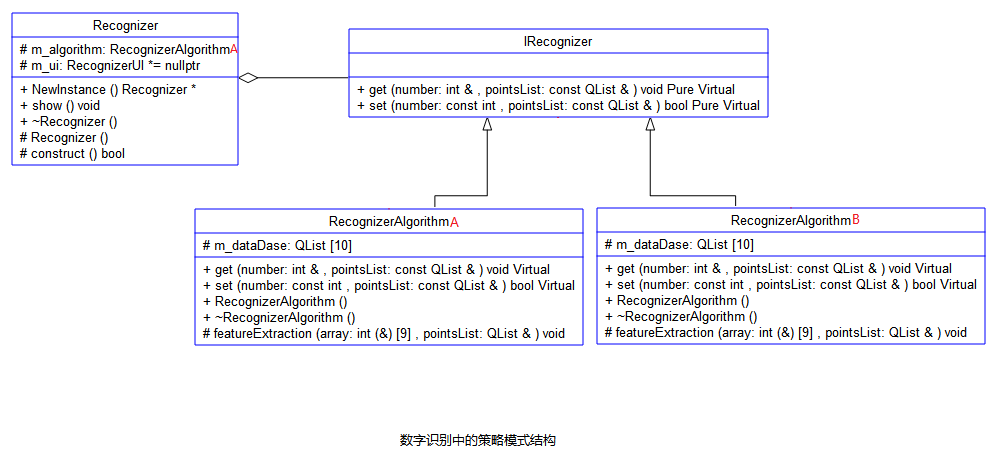
数字识别类（Recognizer）、界面类 (RecognizerUI )、均使用二阶构造模式，防止异常的产生,下面是Recognizer类的中的二阶构造部分。



**2、策略者模式**

策略模式用于算法的自由切换和扩展，对应于解决某一问题的一个算法族，允许用户从该算法族中任意选择一个算法解决问题。为了方便算法族中的不同算法在使用中具有一致性，在策略模式中会提供一个抽象层来声明公共接口

**数字识别中的策略者模式**



**Recognizer**：是使用算法的角色，可以在解决不同具体的问题时实例化不同的具体识别算法类对象

**IRecognizer**：声明算法的方法，抽象层的设计使Recognizer类可以无差别的调用不同的具体算法；

**RecognizerAlgorithmA**和**RecognizerAlgorithmB**：实现具体的一种算法，当前项目中只使用了一种算法。

使用策略者模式的优点：系统需要在一个算法族中动态选择一种算法，可以将这些算法封装到多个具体算法类中，这些算法类都有共同的基类，即可以通过一个统一的接口调用任意一个算法，客户端可以使用任意一个算法，当前使用“向量的空间距离值”算法，之后也可以扩展KNN等识别算法，而只需要很小改动即可。

**四、函数调用流程图及UML类图**

