#### 指标库

指标库用于分析病人的属性，状况等。例如有无吸烟，血压多少这些都是指标库中的内容。

指标库分为了，指标项目和标准指标两块。

**指标项目：**

指标项目为树形结构，例如：

是否吸烟

是

吸烟年龄

每天吸烟支数

否

特定：

1. **是否吸烟**为一项目，**吸烟年龄**和**每天吸烟支数**都是它的子项目，
2. **吸烟年龄**和**每天吸烟支数**都依赖与**是吸烟**这个情况。
3. **是否吸烟**为选择项目，有标准值（标准值是key，value形式，key是有意义并且通用的）提供选择；**吸烟年龄**则是输入的一个数字

**标准指标：**

标准指标是指一系列key，value形式的标签，分为完整和非完整两种；

**1）完整标准：**

表示完整的，有意义，能够独立使用的标准，例如有吸烟，该标准用于直接标记病人情况和属性。

**2）非完整标准：**

表示非完整，必须与其他（指标项目）结合才能使用的标准，例如有，无，与是否吸烟结合能够产生有吸烟这个完整的标准。

两种标准其实是共同的，或者说是一致的，例如当创建是否吸烟和有，无两个可选项时候，则实际创建了两个完整标准：有吸烟，无吸烟

有吸烟 = 是否吸烟 + 有

无吸烟 = 是否吸烟 + 无

这里两种标准的指定实际是用一个值还是两个值去标示唯一的一件事的问题，在这个设计问题上，完整标准有利于直接与病人直接关联，更效率简单的匹配。而非完整标准+项目的方式则提供了内部更好的扩展。

在代码 com.paladin.health.index下已经抽象出了这些实体类，并且在数据库中建立了相应的index开头的表。

TO DO：

1）页面输入标准库

2）数据库对象转换为充血模式对象

3）完善

#### 分析库

我们使用框架表示方法去表示知识（就是根据固定规律去匹配，因为疾病分析与疾病健康处方逻辑相对固定）。

**疾病框架：**

**1）槽位：**

每个疾病拥有多个槽位，每个槽位带有一系列侧写，这些侧写用于匹配是否病人可能有这种疾病。槽位最终应该返回一个匹配程度类似的值，用于系统判定

例如症状槽位，侧写则是该疾病的一些症状。

指标槽位，侧写则是该疾病的一些指标，例如血压多少等

指标槽位的权重应该是高于症状槽位。

1. **值：**

为该疾病相应的健康处方，如果槽位匹配程度达到一定，则可以判定出病人适用该健康处方。

疾病框架也是树形结构的，每个疾病可以继承其他疾病的症状，指标和健康处方，并且扩展他们

**代码，数据库待建**