## 黑盒测试

黑盒测试其实就是功能测试，所谓的黑盒可以理解为，将程序作为一个不能打开的黑盒子在不考虑程序内部结构和内部特性的情况下， 在程序接口进行测试。，（哎~你看不到但是我能检测哦），那么它检测了什么呢？

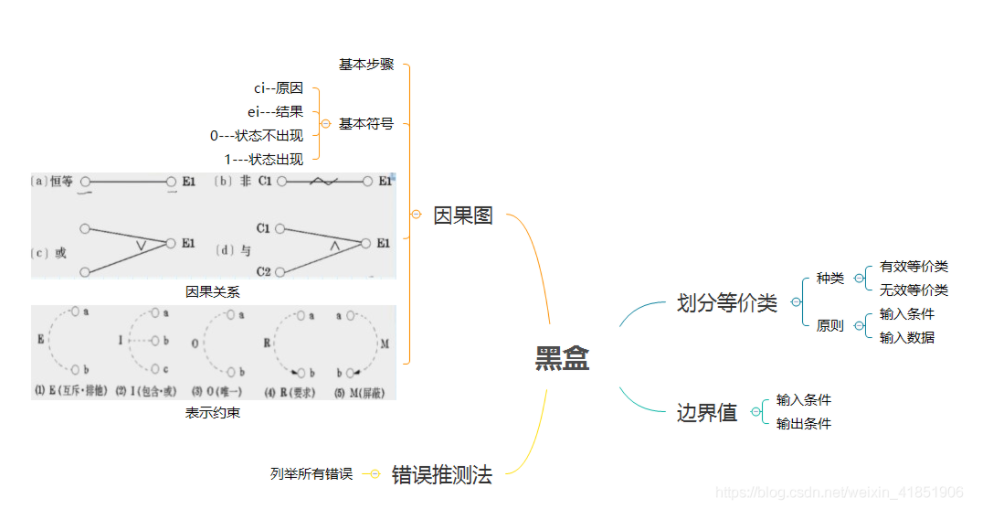
1：检查程序的功能是否符合它的功能说明—需求规格说明书（功能测试/数据驱动）

2：经检查程序的接口

3：对比测试单元的功能

简单说明：

黑盒测试就是在不考虑程序内部结构和特性的情况下，以用户的角度，从输入输出数据的对应关系进行测试。所以如果本身设计/规格说明存在问题，黑盒测试是发现不了的，也就是说它可以判断程序是否有错却不能判断哪里有问题。



划分等价类

- 等价类是什么？

等价类就是输入域的子集合。

- 划分

1.有效等价类

对于程序的规格说明是合理的，有意义的输入数据构成的集合，可以用来检测程序是否实现了规格说明中的性能。

2.无效等价类

与有效等价类相反，对于程序的规格是不合理的，有无意义的输入数据构成的集合。

- 划分原则

1.输入的条件规定了取值范围和值的个数—可以确定一个有效/两个无效等价类

2.规定了输入的值得集合的条件，确定—一个有效/无效等价类

3.输入条件是一个布尔值–什么是布尔值呢？ true/false–确定—一个无效/一个有效等价类

4.规定了输入数据的一组值，对每一个值进行分别处理，确定—n有效/一个无效等价类

5.规定了输入数据遵循的规则的情况下—确定一个/n个无效等价类

边界值

- 边界值是什么？为什么测试边界值

错误发生在输入/输出范围边界上的概率大，也就是说测试边界值是很有必要的，边界值就像是事故频发地带一样。对于这些特殊区域要给予相当大的关注，那么什么是边界值呢，举一个小例子：3<=x+1<=5 x的取值范围：2<=x<=4 边界值则为2/4

- 原则

1.输入条件规定了值的范围，则取达到此范围边界的值，以及刚刚超越此范围边界的值

2.输入条件规定了取值的数量，则比最小个数少一，比最大个数多一

3.输出条件同上

4.输入/输出域是一个有序集合，则选择集合中第一个和最后一个元素进行测试

5.程序使用了颞部数据结构则选择此内部数据结构的边界上的值

6.依据规格说明进行分析，找到其他可能的边界条件

- 与等价类的区别

1.边界值就是边界的值而不是随意选取的

2.边界值分析输入/输出条件都考虑了

因果图法（逻辑模型）

- 基本小点

1.用来描述多种条件的组合

2.最终生成判定表

- 生成测试用例步骤

1.分析原因，结果并赋予标识符

2.找出原因、结果之间的关系

3.用记号标出约束/限制条件

4.将因果图转化为判定表

5.将判定表中的每一列拿出来进行测试