



黑盒测试

张程

Email: bootan@cqu.edu.cn

QQ:80463125

黑盒测试



黑盒测试对程序的功能性测试要求

- 每个软件特性必须被一个测试用例或一个被认可的异常所覆盖
 - 用一系列真实的数据类型和数据值运行，测试超负荷及其他“最坏情况”的结果
 - 用假想的数据类型和数据值运行，测试排斥不规则输入的能力
 - 对影响性能的关键模块，如基本算法、精度、时间、容量等是否正常
- 用数据类型和数据值的最小集测试



黑盒测试的优点

- 黑盒测试与软件具体实现无关，所以如果软件实现发生了变化，测试用例仍然可以使用；
- 设计黑盒测试用例可以和软件实现同时进行，因此可以压缩项目总的开发时间
- 从产品功能角度测试可以最大程度满足用户的需求
- 相同动作可重复执行，最枯燥的部分可由机器完成。
- 依据测试用例针对性地找寻问题，定位更为准确，容易生成测试数据。
- 将测试直接和程序/系统要完成的操作相关联。



黑盒测试的缺点

- 代码得不到测试
- 如果规格说明设计有误，很难发现
- 测试不能充分的进行
- 结果取决于测试用例的设计



黑盒测试对测试工作的意义

- 黑盒测试有助于对被测产品进行总体功能的需求进行验证;
- 从测试管理来说, 黑盒测试是非常方便的, 不需要对代码进行测试管理;
- 黑盒测试是把所有可能的输入都作为测试数据使用的, 容易查出程序中的错误。



黑盒测试的原则

- 根据软件规格说明书设计测试用例, 规格说明书的正确性是至关重要的。
- 有针对性的地找问题, 并且正确定位等价类
- 功能是否有缺陷或错误现象?
- 根据测试的重要性来确定测试等级和测试重点, 减少程序可能出现的缺陷
- 在接口处, 输入的信息是否能正确接受? 接受后能否输出正确的结果
- 认真选择测试策略, 尽可能发现程序的数据结构错误或外部信息访问错误, 站在用户立场上进行测试



黑盒测试的方法-等价类划分法

- 等价类划分法
 - 一种典型的、重要的黑盒测试方法, 它将程序所有可能的输入数据 (有效的和无效的) 划分成若干个等价类。然后从每个部分中选取具有代表性的数据当作测试用例进行合理的分类, 测试用例由有效等价类和无效等价类的代表组成, 从而保证测试用例具有完整性和代表性。
 - 等价类划分为有效等价类和无效等价类





黑盒测试的方法-等价类划分法

■ 等价类划分法

- 包括：划分等价类的要求、划分等价类的方法、划分等价类应注意的原则、等价类划分法的建立
- 等价类表的建立如下表所示：

输入条件	有效等价类	无效等价类
-	-	-
-	-	-



黑盒测试的方法-等价类划分法

■ 划分等价类的原则

- 如果输入条件规定了取值范围，或值的个数，则可确立一个有效等价类和两个无效等价类。
 - 如，在程序规格说明中对输入条件有一句话：“..... 项数可以从1到999”
则有效等价类是“ $1 \leq \text{项数} \leq 999$ ”
两个无效等价类是“项数 < 1 ”或“项数 > 999 ”
- 如果输入条件规定了输入值的集合，或者是规定了“必须如何”的条件，这时可确立一个有效等价类和一个无效等价类。
 - 例如，在Pascal语言中对变量标识符规定为“以字母打头的.....串”。那么所有以字母打头的构成有效等价类，而不在集合内（不以字母打头）的归于无效等价类。
- 如果输入条件是一个布尔量，则可以确定一个有效等价类和一个无效等价类。



黑盒测试的方法-等价类划分法

■ 划分等价类的原则

- 如果规定了输入数据的一组值，而且程序要对每个输入值分别进行处理。这时可为每一个输入值确立一个有效等价类，此外针对这组值确立一个无效等价类，它是所有不允许的输入值的集合。
 - 例如，在教师上岗方案中规定对教授、副教授、讲师和助教分别计算分数，做相应的处理。
因此可以确定4个有效等价类为教授、副教授、讲师和助教，一个无效等价类，它是所有不符合以上身分的人员的输入值的集合
- 如果规定了输入数据必须遵守的规则，则可以确立一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。
 - 例如，Pascal语言规定“一个语句必须以分号‘;’结束”。这时可以确定一个有效等价类“以‘;’结束”，若干个无效等价类“以‘.’结束”、“以‘,’结束”、“以‘ ’结束”、“以LF结束”。



黑盒测试的方法

■ 等价类划分法

- 示例
- “用户登录”的功能需求如下：

用户名为字母、数字的组合，且字母不区分大小写；
用户名只能以字母开头；
用户名长度为8到16位；
不能为空、空格和特殊字符；
密码与用户名要求一致；

输入条件	有效等价类	无效等价类
用户名字符组合	①字母（a~z或A~Z）和数字（0~9）组合	②特殊字符 ③纯数字组合 ④纯字母组合
	⑤[8~16]数字字母组合	⑥小于8位的数字字母组合 ⑦大于16位的数字字母组合 ⑧用户名为空 ⑨用户名为空格
用户名长度	⑩字母开头的用户名	⑪数字开头的用户名 ⑫其他字符开头的用户名
用户名格式	⑬字母（a~z或A~Z）和数字（0~9）组合	⑭特殊字符 ⑮纯数字组合 ⑯纯字母组合
密码字符组合	⑰[8~16]数字字母组合	⑱小于8位的数字字母组合 ⑲大于16位的数字字母组合 ⑳密码为空 ㉑密码为空格
密码长度	㉒字母开头的密码	㉓数字开头的密码 ㉔其他字符开头的密码
密码格式	22字母开头的密码	



黑盒测试的方法-等价类划分法

用例ID	功能点	测试输入	预期结果	实际结果	覆盖的等价类
YHDL-0002	用户登录	用户名: adf@1234; 密码: zwb9641asdf1234;	登录失败	登录失败	②
YHDL-0003	用户登录	用户名: 1234567890; 密码: zwb9641asdf1234;	登录失败	登录失败	③
YHDL-0004	用户登录	用户名: qwerasdfzxcv; 密码: zwb9641asdf1234;	登录失败	登录失败	④
YHDL-0005	用户登录	用户名: asd1234; 密码: zwb9641asdf1234;	登录失败	登录失败	⑥
YHDL-0006	用户登录	用户名: qwer1234asdf1234; 密码: zwb9641asdf1234;	登录失败	登录失败	⑦
YHDL-0007	用户登录	用户名: 为空; 密码: zwb9641asdf1234;	登录失败	登录失败	⑧
YHDL-0008	用户登录	用户名: (空格); 密码: zwb9641asdf1234;	登录失败	登录失败	⑨
YHDL-0009	用户登录	用户名: 1234qwer; 密码: zwb9641asdf1234;	登录失败	登录失败	⑪
YHDL-0010	用户登录	用户名: 或1234qwer; 密码: zwb9641asdf1234;	登录失败	登录失败	⑫



等价分类法-示例2

■ 三角形问题:

输入三个100以内的整数a,b和c分别作为三角形的3条边, 通过程序判断由这3条边构成的三角形的类型是: 等边三角形, 等腰三角形, 一般三角形或非三角形。

隐含规则	有效等价类	无效等价类
$1 <= a <= 100$ $a < b + c$	整数	一边为非整数 二边为非整数 三边为非整数
- 等价类	3个数	只有一条边 只有二条边 只有三条边
	$1 <= a <= 100$ $1 <= b <= 100$ $1 <= c <= 100$,	一边<1 二边<1 三边<1 一边>100 二边>100 三边>100



黑盒测试的方法-边界值分析法

■ 边界值分析法

- 用于对输入或输出的边界值进行测试的一种典型、重要的黑盒测试方法。在测试过程中, 边界值分析法是作为对等价类划分法的补充, 专注于每个等价类的边界值, 区别是在等价类中随机选取一个测试点
- 边界条件: 软件在能力发挥到极限的情况下能够正常运行, 则在普通情况下运行一般也不会存在问题
- 次边界条件: 普通边界条件一般在产品说明书中有明确定义, 或者在使用软件的过程中能够的值。而有些边界条件在软件内部, 软件使用者基本无法得知, 但软件测试人员必须考虑。此类边界条件成为次边界条件或内部边界条件
- 其他边界条件: 用户在输入框中输入登录信息或填写其他信息时, 可能并没有输入数据, 即直接回车。产品说明书一般会把这些情况忽略, 但在现实生活中却时常发生。完善的软件通常会将输入内容默认为合法边界内的最小值或合法边界内的某个合理值, 当没有值时直接返回提示信息, 如“不能为空”等。



边界值分析

■ 测试用例的原则

- 如果输入条件规定了值的范围, 则应取刚刚到达这个范围边界的值, 以及刚刚超过这个范围边界的值作为测试输入数据。
 - 例如, 某数据的取值范围为-1.0~1.0, 测试数据可取-1.0、1.0, 以及-1.1、1.1
- 如果输入条件规定了值的个数, 则应取最大个数、最小个数, 比最大个数多1, 比最小个数少1的数作为测试输入数据。
 - 例如, 某文件有255个记录, 测试数据可取1、255, 以及0、256。
- 根据规格说明和每个输出条件, 使用原则(1)。
 - 例如, 研究生录取分数范围84~150, 测试数据可取84、150, 以及83、151。
- 根据规格说明和每个输出条件, 使用原则(2)。
 - 例如, 研究生录取人数34人, 测试数据可取1、34、以及0、35。
- 如果程序的规格说明给出的输入域或输出域是有序集合(如有序表), 则选取集合的第一个元素和最后一个元素作为测试用例。
 - 例如, 学生文件的学生记录按学号存放, 班上总共30人, 测试数据可取第1、第30个学生。
- 如果程序中使用了一个内部数据结构, 则应选择此数据结构的边界上的值作为测试用例。



黑盒测试的方法-边界值分析法

■ 边界值分析法

- 示例

- “用户登录”的功能需求如下:

用户名为字母、数字的组合, 且字母不区分大小写;

用户名只能以字母开头;

用户名长度为8到16位;

不能为空、空格和特殊字符;

密码与用户名要求一致;

输入条件	边界值	取值	编号
用户名组合	字母 (a—z或A—Z)	a	①
		z	②
	数字 (0—9)	0	③
		9	④
用户名长度	[8—16]位	NULL	⑤
		7位	⑥
		8位	⑦
		16位	⑧
		17位	⑨
密码组合	字母 (a—z或A—Z)	a	⑩
		z	⑪
	数字 (0—9)	0	⑫
		9	⑬
密码长度	[8—16]位	0 (空)	⑭
		7位	⑮
		8位	⑯
		16位	⑰
		17位	⑱



黑盒测试的方法

■ 边界值分析法

用例ID	功能点	测试输入	预期结果	实际结果	覆盖的边界值
YHDL-bjz0001	用户登录	用户名: s1234567; 密码: cwb9641asdf1234	登录成功	登录成功	①
YHDL-bjz0002	用户登录	用户名: z1234567; 密码: cwb9641asdf1234	登录成功	登录成功	②
YHDL-bjz0003	用户登录	用户名: asdfzxc0; 密码: cwb9641asdf1234	登录成功	登录成功	③
YHDL-bjz0004	用户登录	用户名: asdfzxc9; 密码: cwb9641asdf1234	登录成功	登录成功	④
YHDL-bjz0005	用户登录	用户名: Null; 密码: cwb9641asdf1234	登录失败	登录失败	⑤
YHDL-bjz0006	用户登录	用户名: s123456; 密码: cwb9641asdf1234	登录失败	登录失败	⑥
YHDL-bjz0007	用户登录	用户名: s1234567; 密码: cwb9641asdf1234	登录成功	登录成功	⑦
YHDL-bjz0008	用户登录	用户名: asdfzxcv12345678; 密码: cwb9641asdf1234	登录成功	登录成功	⑧
YHDL-bjz0009	用户登录	用户名: asdfzxcv123456789; 密码: cwb9641asdf1234	登录失败	登录失败	⑨
YHDL-bjz0010	用户登录	用户名: asdf4567; 密码: s1234567	登录成功	登录成功	⑩
YHDL-bjz0011	用户登录	用户名: asdf1234; 密码: z1234567	登录成功	登录成功	⑪



黑盒测试的方法-组合测试用例设计技术

■ 无论等价类划分法还是边界值法, 都假定程序的各个输入变量是完全独立的。但在实际程序中, 更多的情况是各个输入变量的组合共同导致了程序的输出。

■ 大量的组合测试实验结果表明, 约20%-40%的软件故障时由单个参数引起的, 约70%的软件故障时由单个参数引起或两个参数的相互作用引起的, 而只有20%左右的软件故障时由3个或3个以上的参数相互作用引起的。说明组合测试具有非常重要的应用价值



组合测试用例设计技术-全面测试

■ 全面测试需要对所有输入的各个取值之间的各种组合情形均进行相应的测试。对于软件测试而言, 假设被测功能有m个输入, 且每个输入有多个离散但有限的取值N1、N2.....Nm(其中Ni的值可以不等, $1 \leq i \leq m$), 为了覆盖输入参数的全部取值组合, 需要 $N1 * N2 * ... * Nm$ 个测试用例。当测试问题可以描述为一组参数, 且每个参数有多个值, 致使可能组合的参数值的总数大到测试不可行时, 称为组合爆炸

■ 以共享单车扫码功能为例 (如图6-3所示), 考察全面测试时测试用例的数量。对扫描二维码功能有影响的条件有: 网络情况、光照、距离和二维码的完整性。



影响条件	等价类
网络	4G、3G、无线网络、无网络
光照	强、弱、正常
距离	0.3m-1.5m、<0.3m、>1.5m
二维码完整性	正常、缺少一个定位点、缺少1/2的非定位点



组合测试用例设计技术-全面测试

- 优点
 - 各个参数的所有取值组合都得到测试，可以发现任何与参数组合相关的错误
- 缺点
 - 当参数数量与取值个数都较大时，所需要的测试用例数量将非常庞大，这对于资源有限的软件测试而言，是不可行的
 - 有时很多组合实际上对于被测软件而言，排错率相当低
- 全面测试是最完备的组合测试模型，但对于以人工测试为主的软件测试，全面测试的可行性大打折扣



组合测试用例设计技术-单因素覆盖

- 若测试用例集中的数据包含了每个因素的所有取值，则称测试用例集符合单因素覆盖标准。
- 设有以下需求规格说明要求：一个程序接受三个输入值A、B、C，A从集合{A1, A2, A3}中取值，B从集合{B1, B2, B3}中取值，C从集合{C1, C2, C3, C4}中取值，则测试用例集的规模为4
- 分别从A,B,C三个取值集取值，形成下表的组合。这样，测试用例集中的数据包含了ABC每个因素的所有取值，形成了单因素覆盖。
- 单因素覆盖只强调测试用例集要覆盖每个因素的取值
- 通过组合法设计测试用例，必须结合业务实际需求来组合数据，否则设计的测试用例必定漏洞百出

编号	测试用例
T1	A1 B2 C1
T2	A2 B3 C3
T3	A3 B1 C2
T4	A1 B2 C4



- 某系统的合同详情页面中，收益日期的显示需满足如下规则：
 - 收益日期生成规则：佣金（中介费）、押金、首期房租到账、
 - 合同审批通过、物业交割单审批通过后，显示收益日期；
 - 收益日期取最晚审核通过的日期。
- 收益日期生成规则由5个因素组成，可将每个因素从业务需求出发进行等价类划分：
 - 佣金：{收齐，未收}
 - 押金：{收齐，未收}
 - 首期房租：{收齐，未收}
 - 合同审批：{通过，不通过}
 - 物业单审批：{通过，不通过}

当前位置：承租合同详情

承租合同编号：	CZ100000	承租租金币种：	元
收益日期：		签约方式：	纸质签约

房屋信息

房源编号：	FY1000150	房屋类型：	两室 1厅 1卫		
客户居住类型：	整租	区域：	南滨区		
产权证地址：	*****	查看	产权证地址：	*****	查看

租客信息

租客姓名：	张三	租客证件：	1122345678908
租客电话：	15621213456	租客类型：	正常客户
房东/房东租客：	否	联系人姓名：	李四
联系人电话：	13422190066	户籍地址：	北京市海淀区****



序号	测试用例描述
T1	三金收齐+合同审核通过+物业交割单审批通过→产生收益日期，且均取其最晚审核通过的日期
T2	三金收齐+合同审核通过+物业交割单提交审核(即审核未通过)→无收益日期
T3	三金收齐+合同提交审核(即审核未通过)+物业交割单审批通过→无收益日期
T4	佣金、押金收齐,首期房租未收+合同审核通过+物业交割单审批通过→无收益日期
T5	押金、首期房租收齐,佣金未收+合同审核通过+物业交割单审批通过→无收益日期
T6	佣金、首期房租收齐,押金未收+合同审核通过+物业交割单审批通过→无收益日期
T7	佣金为0，押金、首期房租收齐+合同审核通过+物业交割单审批通过→有收益日期
T8	押金为0，佣金、首期房租收齐+合同审核通过+物业交割单审批通过→有收益日期
T9	佣金、押金收齐、首期房租部分收款+合同审核通过+物业交割单审批通过→无收益日期



组合测试用例设计技术-两两组合

- 使用两两组合法设计测试用例会进一步减少测试用例数量，并且该方法也能够达到全面覆盖参数取值两两组合的测试需要。
- 设有以下需求规格说明要求：一个程序接受三个输入值x1、x2、x3，x1可取值a1、a2、a3，x2可取值b1、b2，x3可取值c1、c2，x1、x2、x3所有取值的两两组合为：
 - (a1, b1), (a1, b2), (a2, b1), (a2, b2), (a3, b1), (a3, b2);
 - (a1, c1), (a1, c2), (a2, c1), (a2, c2), (a3, c1), (a3, c2);
 - (b1, c1), (b1, c2), (b2, c1), (b2, c2)。

测试用例序号	x ₁	x ₂	x ₃
1	a ₁	b ₁	c ₁
2	a ₁	b ₂	c ₂
3	a ₂	b ₁	c ₁
4	a ₂	b ₂	c ₂
5	a ₃	b ₁	c ₂
6	a ₃	b ₂	c ₁



组合测试用例设计技术-具有约束关系的组合测试

- 在实际问题中，某些参数之间的取值是有一定的约束关系的，这反映在组合测试问题上就是测试用例集中的某些组合是无效的或没有意义的。
- 设有以下需求规格说明要求：一个程序接受三个输入值x1、x2、x3，输入参数表及其取值表中附加了约束条件，如表6-17所示。取值约束条件共3条：
- 约束条件<a2, ~b1>：表示在一个测试用例中，当参数x1取a2时，参数x2就不能取b1；
- 约束条件<c1, a1>：表示参数x3取c1时，参数x1只能取a1；
- 条件<b2, c2>：表示在测试用例中，当参数x2取b2时，参数x3只能取值为c2。

x ₁	x ₂	x ₃
a ₁	b ₁	c ₁
a ₂	b ₂	c ₂
a ₃		
约束条件：		
(1) <a ₂ , ~b ₁ >		
(2) <c ₁ , a ₁ >		
(3) <b ₂ , c ₂ >		



- 软约束：软约束又称非强制性约束，测试用例集中是否出现软约束不会影响测试用例集的错误检测能力，如果在生成测试用例时考虑这种约束，就会在保证错误检测能力的前提下进一步减少测试用例集的大小，从而降低测试成本。
- 硬约束：硬约束又称强制性约束，一般情况下是不允许测试用例集出现硬约束的，否则将会影响到测试用例集的错误检测能力。
- 以注册功能为例，说明这种设计测试用例方法的应用。在实际应用中，身份证号与性别存在2个约束关系：
 - <身份证号偶数第二位为奇数，性别男>
 - <身份证号偶数第二位为偶数，性别女>



填写基本信息

姓名:

性别: ☒ 男 ☐ 女

身份证号:

密码:

验证码:

姓名	性别	身份证号	密码	验证码
John	男	110101199003077635	QQabc123	60s内正常验证码
U0001	女	110101199003078507	123abA	

姓名	性别	身份证号	密码	验证码
U0001	男	110101199003077635	123abA	60s内正常验证码
John	女	110101199003078507	123abA	60s内正常验证码
U0001	女	110101199003078507	QQabc123	60s内正常验证码
John	男	110101199003077635	QQabc123	60s内正常验证码



黑盒测试的方法-因果图法

■ 因果图法

- 一种适合于描述对于多种输入条件组合的测试方法，根据输入条件的组合、约束关系和输出条件的因果关系，分析输入条件的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件到输出结果的各种组合情况。因果图法一般和判定表结合使用通过映射同时发生相互影响的多个输入来确定判定条件。因果图法最终生成的就是判定表。它适合于检查程序输入条件的各种组合情况
- 因果图法关系符号 C代表原因，E代表结果
- 因果图中使用了简单的逻辑符号，以直线联接左右结点。左结点表示输入状态（或称原因），右结点表示输出状态（或称结果）
- C_i 表示原因，通常置于图的左部； E_i 表示结果，通常在图的右部。 C_i 和 E_i 均可取值0或1，0表示某状态不出现，1表示某状态出现

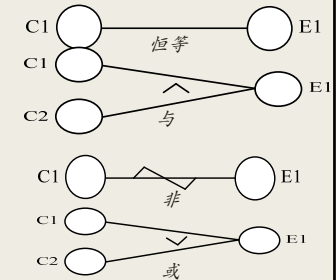


黑盒测试的方法-因果图法

■ 基本符号

■ 输入条件与输出结果之间的关系

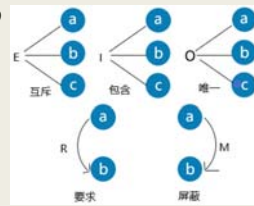
- 恒等：若原因出现，则结果出现；若原因不出现，则结果也不出现。例如：
 - 若 $C1=1$ ，则 $E1=1$ ；若 $C1=0$ ，则 $E1=0$ 。
- 非：若原因出现，则结果不出现；若原因不出现，则结果出现。例如：
 - 若 $C1=1$ ，则 $E1=0$ ；若 $C1=0$ ，则 $E1=1$ 。
- 或：若几个原因中有一个出现，则结果出现；若几个原因都不出现则结果不出现。例如：
 - 若 $C1=1$ 或 $C2=1$ ，则 $E1=1$ ；若 $C1=C2=0$ ，则 $E1=0$ 。
- 与：若几个原因都出现，结果才出现；若其中有一个原因不出现，则结果不出现。例如：
 - 若 $C1=C2=1$ ，则 $E1=1$ ；若 $C1=0$ 或 $C2=0$ ，则 $E1=0$ 。



黑盒测试的方法-因果图法

■ 输入或输出的约束关系

- 输入状态相互之间还可能某些依赖关系，称为约束。例如，某些输入条件本身不可能同时出现。输出状态之间也往往存在约束。在因果图中用特定的符号标明这些约束
- E约束符号（互斥）：a和b中至多有一个可能为1，即a和b不能同时为1
- I约束符号（包含）：a、b和c中至少有一个必须是1，即a、b和c不能同时为0
- O约束（唯一）：a和b必须有一个且仅有一个为1
- R约束（要求）：a是1时，b必须是1，即a为1时，b不能为0
- M约束（强制）：若结果a为1，则结果b强制为0。



黑盒测试的方法-因果图法

■ 特点

- 考虑输入条件的组合关系
- 考虑输出条件对输入条件的信赖关系，即因果关系
- 测试用例发现错误的效率高
- 可检查出功能说明书中的某些不一致或遗漏
- 适合于检查程序输入条件和各种组合情况



黑盒测试的方法-因果图法

- 导出测试用例一般要经过以下几个步骤：
 - 分割功能说明书。对规模较大的程序，将其划分为若干个部分，然后对每个部分使用因果图法
 - 分析软件规格说明的描述中那些是原因，那些是结果。原因是输入或输入条件的等价类，结果是输出条件。并给每个原因和结果赋予一个标示符，根据这些关系，画出因果图。
 - 因果图上用一些记号表明约束条件或限制条件。
 - 对需求加以分析并把它们表示为因果图之间的关系图。
 - 把因果图转换成判定表。
 - 根据判定表的每一列作为依据，设计测试用例。



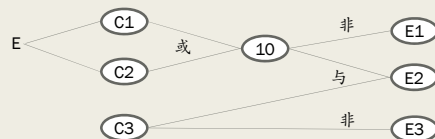
黑盒测试的方法-因果图法

- 示例：某软件规格说明书规定：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。
- 测试设计步骤如下：
 - (1) 分析软件规格说明书，找出原因和结果
 - 原因：
 - 第一个字符是#
 - 第一个字符是*
 - 第二个字符是一个数字
 - 结果：
 - 给出信息N
 - 修改文件
 - 给出信息M



黑盒测试的方法-因果图法

- 示例：某软件规格说明书规定：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。
- 测试设计步骤如下：
 - (2) 找出原因与结果之间的因果关系，原因与原因之间的约束关系，画出因果图



黑盒测试的方法-因果图法

- 示例：某软件规格说明书规定：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。
- 测试设计步骤如下：
 - (3) 将因果图转换成决策表。

根据因果图建立起决策表，如表 4-15 所示。

	1	2	3	4	5	6	7	8
条件:								
C ₁	1	1	1	1	0	0	0	0
C ₂	1	1	0	0	1	1	0	0
C ₃	1	0	1	0	1	0	1	0
10			1	1	1	1	0	0
动作:								
e ₁							✓	✓
e ₂			✓		✓			
e ₃				✓		✓		
不可能	✓	✓						



黑盒测试的方法-因果图法

- 示例：某软件规格说明书规定：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。

- 测试设计步骤如下：

（4）根据决策表设计测试用例的输入数据和预期输出。

表中 8 种情况的最左面两列，原因和同时为 1，这是不可能的，故应排除这两种情况，根据该表，我们设计出了 6 个测试用例，如表 4-16 所示。

表 4-16 依据决策表设计的测试用例

测试用例编号	输入数据	预期输出
1	#3	修改文件
2	#A	给出信息 M
3	*6	修改文件
4	*B	给出信息 M
5	A1	给出信息 N
6	GT	给出信息 M 和信息 N



黑盒测试的方法-因果图法

- 示例2

- 有一个饮料自动售货机（处理单价为5角钱）的控制处理软件，它的软件规格说明如下：
- 若投入5角钱的硬币，按下“橙汁”或“啤酒”的按钮，则相应的饮料就送出来。若投入1元钱的硬币，同样也是按“橙汁”或“啤酒”的按钮，则自动售货机在送出相应饮料的同时退回5角钱的硬币。

原因	结果
1：投入1元硬币	21：退还5角硬币
2：投入5角硬币	22：送出“橙汁”
3：按下“橙汁”按钮	23：送出“啤酒”
4：按下“啤酒”按钮	

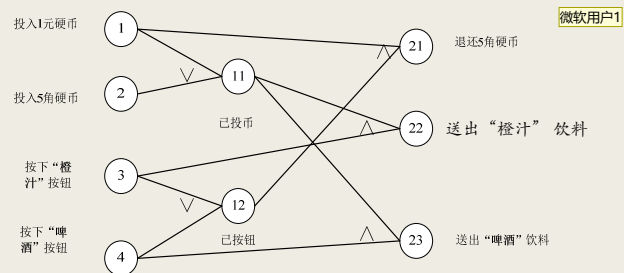


黑盒测试的方法-因果图法

- 画出因果图

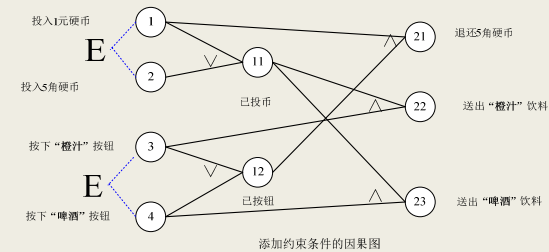
11和22是中间节点

原因	结果
1：投入1元硬币	21：退还5角硬币
2：投入5角硬币	22：送出“橙汁”
3：按下“橙汁”按钮	23：送出“啤酒”
4：按下“啤酒”按钮	



黑盒测试的方法-因果图法

- 施加相关约束





黑盒测试的方法-因果图法

■ 因果图转为判定表

从因果图导出的判定表		1	2	3	4	5	6	7	8
输入	投入1元硬币	1	1	1	0	0	0	0	0
	投入5角硬币	2	0	0	1	1	1	0	0
	按下橙汁	3	1	0	0	1	0	1	0
	按下啤酒	4	0	1	0	0	1	0	1
中间节点	已投币	11	1	1	1	1	1	0	0
	已按钮	12	1	1	0	1	1	0	1
输出	退还5角硬币	21	1	1	0	0	0	0	0
	送出橙汁饮料	22	1	0	0	1	0	0	0
	送出啤酒饮料	23	0	1	0	0	1	0	0



黑盒测试的方法-因果图法

■ 实例分析

■ 公交一卡通自动充值系统，其需求描述如下：

- 系统只接收50元或100元纸币，一次充值只能使用一张纸币，一次充值金额只能为50元或100元。
- 若输入50元纸币，并选择充值50元，完成充值后退卡，提示充值成功。
- 若输入50元纸币，并选择充值100元，提示输入金额不足，并退回50元。
- 若输入100元纸币，并选择充值50元，完成充值后退卡，提示充值成功，找零50元。
- 若输入100元纸币，并选择充值100元，完成充值后退卡，提示充值成功。
- 若输入纸币后在规定时间内不选择充值按钮，退回输入的纸币，并提示错误。
- 若选择充值按钮后不输入纸币，提示错误。



黑盒测试的方法-因果图法

■ 实例分析

■ 条件之间的制约及组合关系

- 输入条件（原因）为：

- ① 投币50元
- ② 投币100元
- ③ 选择充值50元
- ④ 选择充值100元

- 输出（结果）为：

- (a) 完成充值、退卡
- (b) 提示充值成功
- (c) 找零
- (d) 提示错误



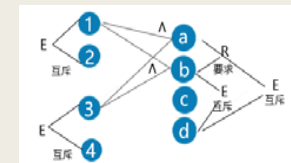
■ 明确所有条件之间的制约关系以及组合关系

- 条件之间的制约关系以及组合关系如图所示。



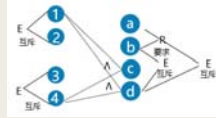
■ 画出因果图

- 为了描述得更清楚，这里把每种情况单独画一个因果图出来。
- 条件①和条件③可以组合，输出a和d的组合，也就是投币50元，充值50元，会输出完成充值、退卡，提示充值成功的结果。其因果图如图所示。

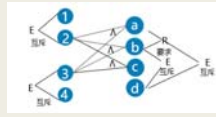




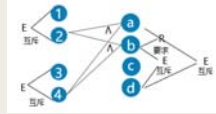
- 条件①和条件④可以组合，输出c和d的组合，也就是投币50元，充值100元，会输出找零、提示错误的结果。其因果图如图所示。



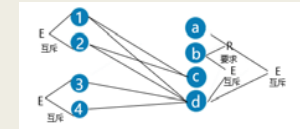
- 条件②和条件③可以组合，输出a、b、c的组合，也就是投币100元，充值50元，会输出找零、完成充值、提示充值成功的结果。其因果图如图所示。



- 条件②和条件④可以组合，输出a和b的组合，也就是投币100元，充值100元，会输出完成充值、退卡，提示充值成功的结果。其因果图如图所示。



- 条件①、②、③、④均可以单独出现，其因果图如图所示。



- 根据因果图，写出判定表

- 根据上面的因果图，写出对应的判定表

		1	2	3	4	5	6	7	8
输入	①投币50元	1	1			1			
	②投币100元			1	1		1		
	③选择充值50元		1	1				1	
	④选择充值100元			1	1				1
输出	(a) 完成充值、退卡	1	1	1					
	(b) 提示充值成功	1	1	1					
	(c) 找零		1	1	1	1			
	(d) 错误提示		1			1	1	1	1



- 根据判定表，写出测试用例

- 根据上面的判定图，写出对应的测试用例

编号	用例说明	预期结果
1	投币50元 选择充值50元	正确充值50元，提示充值成功后退卡
2	投币50元 选择充值100元	系统提示错误并退回50元
3	投币100元 选择充值50元	正确充值50元，提示充值成功后退卡，并找回50元
4	投币100元 选择充值100元	正确充值100元，提示充值成功后退卡
5	投币50元	系统提示错误并退回50元
6	投币100元	系统提示错误并退回100元
7	选择充值50元	系统提示错误
8	选择充值100元	系统提示错误



黑盒测试的方法-判定表驱动测试法

- 判定表驱动测试

- 判定表测试严格，能够将复杂的逻辑关系和多种条件组合的情况表得既具体又明确。针对不同的逻辑条件组合值，分别执行不同的操作。因此，使用判定表能够设计书完整的测试用例集合。判定表一种针对存在条件、动作关系或者因果关系的特性测试的用例设计方法

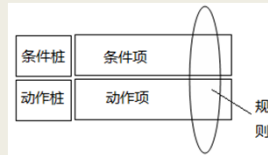


黑盒测试的方法

■ 判定表驱动测试

- 判定表的组成

- 条件桩 (Condition Stub)：列出了问题的所有条件.列出条件的次序没有约束。
- 动作桩 (Action Stub)：列出问题规定可能采取的操作.这些操作的排列顺序无关紧要。
- 条件项 (Condition Entry)：列出条件桩给出的条件并列出所有可能取值。针对条件桩的取值
- 在所有可能情况下的真假值。
- 动作项 (Action Entry)：列出在条件项的各种取值情况下应该采取的动作。
- 规则：任何一个条件组合的特定取值及其相应要执行的操作称为规则。在判定表中贯穿条件项和动作项的一列就是一条规则。有n个条件，每个条件有两个取值（0, 1），故有2n条规则。显然，判定表中列出多少组条件取值，也就有多少条规则。



黑盒测试的方法

■ 判定表驱动测试

- 规则合并：就是有两条或多条规则合并为一条规则。

- 合并条件：

左端：1, 1；合并为右端：1。

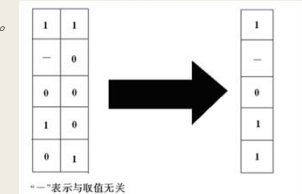
左端：—, 0；合并为右端：—。

左端：0, 0；合并为右端：0。

左端：1, 0；合并为右端：1。

左端：0, 1；合并为右端：1。

- 无关条件项“—”可包含其他条件项取值，具有相同动作的规则可合并，如图所示。



“—”表示与取值无关



黑盒测试的方法-判定表驱动测试法

■ 判定表的建立步骤

- 确定规则的个数加入有n个条件的决策表有2的n次方个规则（每个条件取真、假值）。
- 列出所有的条件桩和动作桩。
- 填入条件项。
- 填入动作项，得到初始判定表。
- 简化判定表，合并相似规则。
- 判定表驱动法的优点和缺点

■ 判定表驱动法的优点：

- 在于它能够把复杂的问题按各种可能的情况一一列举出来,简明而易于理解,也可避免遗漏。在一些数据处理问题当中,某些操作的实施依赖于多个逻辑条件的组合,即:针对不同逻辑条件的组合值,分别执行不同的操作。判定表很适合于处理这类问题。

■ 判定表驱动法的缺点：

- 在于它不能表达重复执行的动作,例如循环结构。



黑盒测试的方法-判定表驱动测试法

■ 判定表驱动测试

■ 示例

■ 在使用判定表驱动法分析时，可以写出以下三个步骤：识别条件和动作、生成判定表、简化判定表

■ 识别条件和动作

条件桩	条件项
用户名为以字母开头长度为8-16位的字母、数字的组合	1: 用户名为以字母开头长度为8-16位的字母、数字组合的字符串 0: 不合法的字符串或空
密码为以字母开头长度为8-16位的字母、数字的组合	1: 密码为以字母开头长度为8-16位的字母、数字组合的字符串 0: 密码为不合法的字符串或空
点击登录按钮	1: 点击登录
不点击登录按钮	2: 不点击登录
动作桩	
登录成功	T表示登录成功
登录失败	F表示登录失败



黑盒测试的方法

- 判定表驱动测试
- 示例
- 生成判定表

条件桩	条件项							
用户名为以字母开头长度为8-16位的字母、数字的组合	1	1	0	0	1	1	0	0
密码为以字母开头长度为8-16位的字母、数字的组合	1	0	1	0	1	0	1	0
点击登录按钮	1	1	1	1	0	0	0	0
动作桩	动作项							
登录成功	T							
登录失败		F	F	F	F	F	F	F



黑盒测试的方法

- 判定表驱动测试
- 示例
- 简化判定表

条件桩	条件项							
用户名为以字母开头长度为8-16位的字母、数字的组合	1	1	0	0	1	1	—	
密码为以字母开头长度为8-16位的字母、数字的组合	1	0	1	0	1	0	1	
点击登录按钮	1	1	1	1	0	0	0	
动作桩	动作项							
登录成功	T							
登录失败		F	F	F	F	F	F	



黑盒测试的方法-判定表驱动测试法

- 判定表 示例2
- 为了获得下一个日期，NextDate函数执行如下操作：
 - 如果输入日期不是当月最后一天，则把day变量的值加1；
 - 如果输入日期是1~11月份中某月的最后一天，则把day变量的值复位为1，month变量的值加1；
 - 如果输入日期是12月的最后一天，则day变量和month变量的值都复位为1，year变量的值加1。
- 关于最后一天的判断：
 - 如果是有31天的月份(1,3,5,7,8,10,12)，day=31为最后一天；
 - 如果是有30天的月份(4,6,9,11)，day=30为最后一天；
 - 如果是有29天的月份(闰年的2月)，day=29为最后一天；
 - 如果是有28天的月份(非闰年的2月)，day=28为最后一天。
- 程序的动作主要是d, m, y的加1或复位，而这些操作又与d, m, y的特征有关。因此可以把d, m, y的加1或复位作为动作桩，d, m, y的特征作为条件桩。



黑盒测试的方法

- 判定表 示例2
- 列出所有的条件桩和行动桩
- 修改条件桩，简化表述

等价类 M1={月份:每月有30天} M2={月份:每月有31天,12月除外} M3={月份:此月是12月} M4={月份:此月是2月} D1={日期:1<=日期<=27} D2={日期:日期=28} D3={日期:日期=29} D4={日期:日期=30} D5={日期:日期=31} Y1={年:年是闰年} Y2={年:年是平年}	条件桩: C1:月份在{M1,M2,M3,M4}中之一 C2:日期在{D1,D2,D3,D4,D5}中之一 C3:年在{Y1,Y2}中之一	动作桩: A1:不可能 A2:日期增1 A3:日期复位(置1) A4:月份增1 A5:月份复位(置1) A6:年增1
---	--	--



黑盒测试的方法

■ 判定表 示例2

规则 桩	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
C1:月份在	M1	M1	M1	M1	M1	M2	M2	M2	M2	M2	M3	M3	M3	M3	M3	M4	M4	M4	M4	M4	M4	M4
C2:日在	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5	D1	D2	D2	D3	D3	D4	D5
C3:年在	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y1	Y2	Y1	Y2	-	-
A1:不可能					√															√	√	√
A2:日增1	√	√	√			√	√	√	√		√	√	√	√		√	√					
A3:日复位				√						√					√			√	√			
A4:月增1				√						√								√	√			
A5:月复位															√							
A6:年增1															√							



黑盒测试的方法

■ 判定表 示例2-简化后

规则 桩	1-3	4	5	6-9	10	11-14	15	16	17	18	19	20	21-22
C1:月份在	M1	M1	M1	M2	M2	M3	M3	M4	M4	M4	M4	M4	M4
C2:日在	D1-D3	D4	D5	D1-D4	D5	D1-D4	D5	D1	D2	D2	D3	D3	D4,D5
C3:年在	-	-	-	-	-	-	-	-	Y1	Y2	Y1	Y2	-
A1:不可能			√									√	√
A2:日增1	√			√		√		√	√				
A3:日复位		√			√		√			√	√		
A4:月增1		√			√					√	√		
A5:月复位							√						
A6:年增1							√						



黑盒测试的方法

■ 判定表 示例2

■ 写出测试用例

用例ID	月份	日期	年	预期输出
1~3	4	12/28/29	2001	2001年4月13/29/30日
4	4	30	2001	2001年5月1日
5	4	31	2001	不可能
6~9	1	15/28/29/30	2001	2001年1月16/29/30/31日
10	1	31	2001	2001年2月1日
11-14	12	15/28/29/30	2001	2001年12月16/29/30/31日
15	12	31	2001	2002年1月1日
16	2	15	2001	2001年2月16日
17	2	28	2004	2004年2月29日
18	2	28	2001	2001年3月1日
19	2	29	2004	2005年3月1日
20	2	29	2001	不可能
21,22	2	30/31	2001	不可能



黑盒测试的方法(练习)-判定表驱动测试法

■ 假设一个中国的航空公司规定:

- 中国去欧美的航线所有座位都有食物供应, 每个座位都可以播放电影。
- 中国去非欧美的国外航线都有食物供应, 只有商务舱可以播放电影。
- 中国国内的航班的商务舱有食物供应, 但是不可以播放电影
- 中国国内的航班经济舱除非飞行时间大于2小时就有食物供应, 但是不可以播放电影。

■ 请用决策表法设计测试用例。



黑盒测试的方法(答案)-判定表驱动测试法

■ 初始条件桩

条件桩:
C1:航线为国外欧美航线
C2:航线为国外非欧美航线
C3:航线为国内航线
C4:仓位为商务舱
C5:仓位为经济舱
C6:飞行时间小于2h
C7:飞行时间大于等于2h

动作桩:
A1:播放电影
A2:食物供应

■ 优化

等价类
M1={航线为国外欧美航线}
M2={航线为国外非欧美航线}
M3={航线为国内航线}
D1={仓位为商务舱}
D2={仓位为经济舱}
Y1={飞行时间小于2h}
Y2={飞行时间大于等于2h}

条件桩:
C1:航线在{M1,M2,M3}中之一
C2:舱位在{D1,D2}中之一
C3:飞行时间在{Y1,Y2}中之一

动作桩:
A1:播放电影
A2:食物供应

黑盒测试的方法 (答案)

■ 生成判定表

规则	1	2	3	4	5	6	7	8	9	10	11	12
C1:航线在	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3
C2:仓位在	D1	D1	D2	D2	D1	D1	D2	D2	D1	D1	D2	D2
C3:飞行时间在	Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2
A1:播放电影	√	√	√	√	√	√						
A2:食物供应	√	√	√	√	√	√	√	√	√	√		√

■ 简化判定表

规则	1	2	3	4	5	6
C1:航线在	M1	M2	M2	M3	M3	M3
C2:仓位在	--	D1	D2	D1	D2	D2
C3:飞行时间在	--	--	--	--	Y1	Y2
A1:播放电影	√	√				
A2:食物供应	√	√	√	√		√



黑盒测试的方法-场景分析法

- 一般把场景分为基本流和备选流。从一个流程开始，通过描述经过的路径来确定业务过程，经过遍历所有的基本流和备选流来完成整个场景。基本流和备选流往往是产品被同一事件的不同触发顺序和处理结果而形成的事件流，事件触发时的情景便形成了场景。通过描述事件的不同触发情景，能够更好的使测试人员理解场景，并设计出合适的测试用例，使测试用例更容易理解和执行
- 场景用来描述流经用例的路径，从用例开始到结束遍历这条路径上所有的基本流和备选流
- 基本流：按照正确的业务流程实现的一条操作路径（模拟正确的操作流程）。
- 备选流：导致程序出现错误的操作流程（模拟错误的操作流程）。



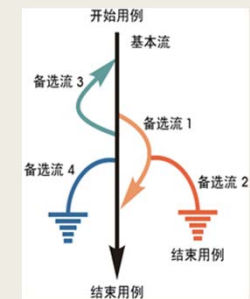
黑盒测试的方法-场景分析法

■ 如何使用场景分析法

- 根据说明，描述出程序的基本流及各项备选流。
- 根据基本流和各项备选流生成不同的场景。
- 对每一个场景生成相应的测试用例。
- 对生成的所有测试用例重新审查，去掉多余的测试用例，
- 确定测试用例后，为每一个测试用例确定测试数据值。

■ 在如图所示的操作流程中，可以确定以下场景：

- 场景1：基本流
- 场景2：基本流 备选流1
- 场景3：基本流 备选流1 备选流2
- 场景4：基本流 备选流3
- 场景5：基本流 备选流3 备选流1
- 场景6：基本流 备选流3 备选流1 备选流2
- 场景7：基本流 备选流4
- 场景8：基本流 备选流3 备选流4





黑盒测试的方法-场景分析法

■ 示例

- **【在线购物系统】** 我们都在当当网或china-pub华章网上书店都订购过书籍，整个订购过程为：用户登录到网站后，进行书籍的选择，当选好自己心仪的书籍后进行订购，这时把所需图书放进购物车，等进行结帐的时候，用户需要登录自己注册的帐号，登录成功后，进行结帐并生成订单，整个购物过程结束。
- 基本流：用户登录到网站，书籍的选择，进行订购，把图书放进购物车，等进行结帐的时候，登录自己的帐号，登录成功后，生成订单
- 备选流：
 - 1帐号不存在
 - 2帐号错误
 - 3密码错误
 - 4无选购书籍
 - 5退出系统



黑盒测试的方法-场景分析法

■ 示例

- 根据基本流和备选流确定场景
 - 场景1-购物成功 基本流
 - 场景2-帐号不存在 基本流，备选流1
 - 场景3-帐号错误 基本流，备选流2
 - 场景4-密码错误 基本流，备选流3
 - 场景5-无选购书籍 基本流，备选流4
- 设计用例
 - 对每个场景都需要确定测试用例，可以采用矩阵或决策表来确定和管理测试用例

测试用例ID	场景/条件	帐号	密码	选购书籍	预期结果
1	购物成功	V	V	V	成功购物
2	帐号不存在	I	N/A	N/A	提示帐号不存在
3	帐号错误	I	V	N/A	提示帐号错误，返回基本流步骤2
4	密码错误	V	I	N/A	提示密码错误，返回基本流步骤3
5	无选购书籍	V	V	I	提示选购数据，返回基本流步骤5



黑盒测试的方法-场景分析法

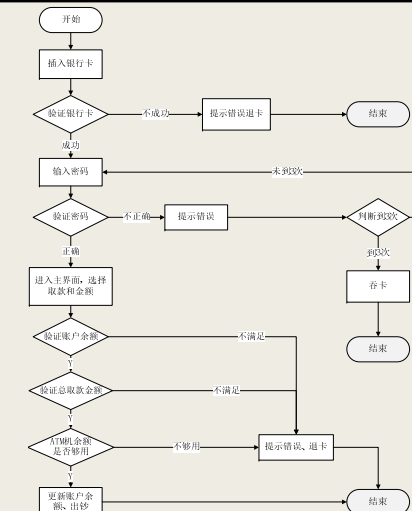
■ ATM机取款的例子



步骤编号	基本流
1	插入银行卡：客户将银行卡插入ATM机的读卡器
2	验证银行卡：ATM机从银行卡的芯片中读取账户代码，并检查它是否属于可以接受的银行卡
3	输入密码：ATM机要求客户输入密码
4	验证密码：确定该密码是否正确
5	进入ATM机主界面：ATM机显示各种操作选项
6	取款并选择金额：客户选择“取款”，并选择取款金额
7	ATM机验证：ATM机验证账户余额、当日总取款金额等是否满足要求，验证ATM机内现金是否够用
8	更新账户余额、出钞：验证成功，更新账户余额，输出现金，提示用户收取现金
9	返回操作主界面




■ ATM机取款的例子



		
场景描述	基本流	备选流
场景1: 成功取款	基本流	
场景2: 银行卡无效	基本流	备选流1
场景3: 密码错误	基本流	备选流2
场景4: 密码3次错误	基本流	备选流3
场景5: 账户余额不足	基本流	备选流4
场景6: 当日总取款金额超出可取限额	基本流	备选流5
场景7: ATM机余额不足	基本流	备选流6


编号	用例描述	预期结果
场景 1	预备条件: ATM机现金余额 5000 元 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 2000 元	
	插入银行卡, 输入正确的密码, 123456, 进入主界面后选择取款 1000 元	① ATM机输出 1000 元, 提示用户取走现金并返回主界面 ② ATM机余额 4000 元 ③ 用户账户余额 1000 元
场景 2	预备条件: ATM机现金余额 4000 元, 准备一张无效银行卡	
	插入无效银行卡	提示该银行卡无效并退卡

		
场景描述	基本流	备选流
场景1: 成功取款	基本流	
场景2: 银行卡无效	基本流	备选流1
场景3: 密码错误	基本流	备选流2
场景4: 密码3次错误	基本流	备选流3
场景5: 账户余额不足	基本流	备选流4
场景6: 当日总取款金额超出可取限额	基本流	备选流5
场景7: ATM机余额不足	基本流	备选流6

场景 3	预备条件: ATM机现金余额 4000 元 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 1000 元。 插入银行卡, 输入错误的密码: 654321	提示密码错误, 并清空密码
场景 4	在场景 3 基础上, 再次输入错误密码: 123123	提示密码错误, 并清空密码
场景 5	预备条件: ATM机现金余额 4000 元 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 1000 元	
	插入银行卡, 输入正确的密码: 123456, 进入主界面后选择取款 2000 元	提示密码错误, 并没收卡片
预备条件: ATM机现金余额 50000 元 (单笔取款最多 2000 元, 一日最多取款 20000 元) 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 30000 元		

		
场景描述	基本流	备选流
场景1: 成功取款	基本流	
场景2: 银行卡无效	基本流	备选流1
场景3: 密码错误	基本流	备选流2
场景4: 密码3次错误	基本流	备选流3
场景5: 账户余额不足	基本流	备选流4
场景6: 当日总取款金额超出可取限额	基本流	备选流5
场景7: ATM机余额不足	基本流	备选流6

场景 6	预备条件: ATM机现金余额 50000 元 (单笔取款最多 2000 元, 一日最多取款 20000 元) 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 30000 元	
	插入银行卡, 输入正确的密码: 123456, 进入主界面后选择取款 2000 元	① ATM机输出 2000 元, 提示用户取走现金并返回主界面 ② ATM机余额 48000 元 ③ 用户账户余额 28000 元
	用户本次累计取款 20000 元	① ATM机余额 30000 元 ② 用户账户余额 10000 元
	再次取款 1000 元	① 提示已达当日最大限额 ② 退卡
场景 7	预备条件: ATM机现金余额 100 元 准备一张有效银行卡, 账号: 955880020013888888, 密码: 123456, 卡内余额 1000 元。	
	插入银行卡, 输入正确的密码: 123456, 进入主界面后选择取款 500 元	提示 ATM机余额不足, 并退卡

		
<h3>黑盒测试的方法(练习)-场景分析法</h3>		
<ul style="list-style-type: none">■ 使用微信钱包中的手机充值, 输入手机号码, 选择话费充值金额, 选择支付方式, 输入支付密码, 支付成功。■ 根据上述描述, 采用场景分析法设计测试用例		



■ 基本流和备选流

基本流	使用微信钱包中的手机充值，输入手机号码，选择话费充值金额，选择支付方式，输入支付密码，支付成功
备选流1	手机号码输入格式错误
备选流2	手机号码输入与要充值的号码不符
备选流3	银行卡余额不足
备选流4	支付密码输入错误



■ 场景设计

场景	路径	
场景1-充值成功	基本流	
场景2-手机号码格式输入错误	基本流	备选流1
场景3-手机号码与充值号码不符	基本流	备选流2
场景4-银行卡账户余额不足	基本流	备选流3
场景5-支付密码输入错误	基本流	备选流4



■ 测试用例设计

用例ID	场景/条件	操作步骤/测试数据	预期结果
1	场景1: 充值成功	1.手机号码输入正确 2.密码输入正确 3.账户余额重铸	充值成功
2	场景2: 手机号码输入错误	输入12356643432334	返回基本步骤1, 重新输入手机号
3	场景3: 手机号码输入与充值号码不符	输入13708322123 (应充值号码为13708322122)	充值成功, 但充值金额未到需充值手机号
4	场景4: 银行卡帐号余额不足	1.手机号码输入正确 2.密码输入正确 3.账户余额不足	提示账户余额不足, 返回选择账户步骤
5	场景5: 支付密码输入错误	1.手机号码输入正确 2.密码输入错误	提示密码输入错误, 请重新输入



黑盒测试的方法-功能图法

■ 功能图法

- 用功能图形象地表示程序的功能说明，由状态迁移图和布尔函数组成，同时需要依靠判定表或因果图表示的逻辑功能，并机械地生成功能图的测试用例，功能图方法是黑盒、白盒混合用例的设计方法
- 功能图模型由状态迁移图和逻辑功能模型两部分构成，程序功能说明包括动态说明和静态说明
- 状态迁移图：用于表示输入数据序列以及相应的输出数据；由输入数据和当前状态决定输出数据和后续状态。
- 逻辑功能模型：用于表示在状态中输入条件和输出条件的对应关系，由输入数据决定输出数据，此模型只适用于描述静态说明。输出数据由输入数据决定
- 动态说明：描述输入数据的次序或转移次序。
- 静态说明：描述输入条件和输出条件之间的对应关系



黑盒测试的方法

■ 功能图法

- 示例
- 该程序为翻译软件，在原文输入框中输入汉语后，选择相应的翻译关系，点击“翻译”可以在输入框中显示对应的内容。按照功能图法方法剖析中讲述的方法，我们逐步进行用例设计，软件界面如图所示



黑盒测试的方法

■ 错误推测法

- 基于以往的经验 and 直觉，参照以往的软件系统出现的错误，推测程序中所有可能存在的各种缺陷和错误，从而有针对性地设计测试用例
- 基本思路是：列举出程序中所有可能的错误和容易发生错误的特殊情况，根据可能出现的错误情况选择测试用例



黑盒测试的方法

■ 特殊值测试

- 特殊值测试就是指定软件中某些特殊值为测试用例而对软件实施的测试。这些特殊值并不是根据某种方法推导出来的，而是根据测试人员的知识和经验得到的。通常情况下，特殊值测试人员都会从过去发生过的失效的事件，或者总会出现问题的情况，或者对于用户来说十分重要的事件中寻找特殊值。

■ 故障猜测法

- 根据经验和直觉猜测软件中可能存在的各种故障，从而有针对性地编写测试这些故障的测试用例。

■ 随机测试

- 对于给定的被测软件系统和软件系统的定义域，按照定义域中样本取值的概率，随机的选择其样本并作为其测试数据的过程称为随机测试。



黑盒测试的方法

■ 错误推测法：基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法

- 时间性测试
 - 提交操作时限
 - 未到达的日期是否可选择
 - 前后时间限制问题
 - 系统时间的调整
- 密码输入框缺陷
 - 明文显示（超级用户）
 - 复制密码，明文显示
 - 一致性
 - 截断：按 Ctrl+V —— 鼠标
 - 限制：新增——修改
- 同时操作问题
 - 在不同机器上登录同一用户
 - 对一条记录在不同机器上进行不同操作（修改、删除）
 - 解决方式一：锁定记录
 - 解决方式二：给出提示信息
 - 一人审核，一人退回
 - 两人修改同一张工单





黑盒测试的方法

■ 常用控件测试

■ 文本框测试

- 数据的内容、长度、类型（注：大小写）、格式（行、日期）、唯一性、空、空格、复制/粘贴+手动、特殊字符、功能键等

1. 输入正常字母或数字。例如在“名称”中输入“工作表 4”，按“确定”，确认在当前的工作表之前建立了一个名称为“工作表 4”的工作表。
2. 输入已存在的工作表名称。例如，已有工作表名为“工作表 4”的工作表，再次输入工作表名称为“工作表 4”，程序应该给出提示。
3. 输入超长字符。例如在“名称”中输入超过允许边界个数的字符，假设最多为 255 个字符，尝试输入 256 个字符，检查程序能否正确处理。
4. 输入默认值，空白，空格。例如在“名称”中输入空格，按“确定”；或者使用默认值，直接按“确定”；或者删除默认值，使内容为空白，按“确定”。
5. 若只允许输入字母，尝试输入数字；若只允许输入数字，尝试输入字母。
6. 利用复制粘贴等操作强制输入程序不允许输入的数据。例如，某文本框只允许输入数字，利用复制粘贴操作，粘贴文本到该文本框，程序应当不允许这种操作。
7. 输入特殊字符集。例如，\NUL，\n 等编程语言中的保留字符。
8. 输入超过文本框长度的字符或输入多行文本，检查输入内容可以正常显示。
9. 输入不符合格式的数据，检查程序能否正确校验。例如，某程序要求输入年月日格式为 yy/mm/dd，实际输入了 yyyy-mm-dd，程序应该给出错误提示。



黑盒测试的方法

■ 常用控件测试

■ 按钮测试

- 按钮功能是否实现（关联）
- 提示信息是否正确（正确、友好、无法恢复时）
- 对于不符合业务背景的输入数据是否有相应的处理



1. 点击按钮正确响应操作。例如点击提交按钮，正确执行报表的添加；点击重置按钮，恢复到初始状态；
2. 对非法的输入或操作给出足够的提示信息。例如输入手机号码不正确时，按“提交”后，系统提示“手机号码不正确”的信息
3. 错误说明应该清楚、明了、恰当，让用户明白错误出处。
4. 对可能造成数据无法恢复的操作必须提供确认信息。给用户放弃选择的机会。例如，删除、关闭、复制、移动等操作。点击删除按钮，系统应该给出提示“是否删除该文件？”；点击关闭按钮，若文件未保存，系统应该给出提示“文件未保存，是否保存该文件？”



黑盒测试的方法

■ 常用控件测试

■ 单选按钮测试

- 各单选按钮功能是否能正确完成
- 单选按钮是否只能选中一个
- 是否有默认被选中的选项
- 功能键（边界）



1. 多个执行同一功能的单选按钮不能同时选中，只能选中其中一个。例如，如图所示的添加留言界面上的选择称呼的单选按钮上，只能选择“先生”或“女士”，如果能同时选中，就是缺陷。
2. 逐一执行每个单选按钮的功能。分别选择了“先生”和“女士”后，保存到数据库中的数据应该相应的分别为“先生”和“女士”。
3. 一组执行同一功能的单选按钮在初始状态时必须有一个默认选中，不能同时为空



黑盒测试的方法

■ 常用控件测试

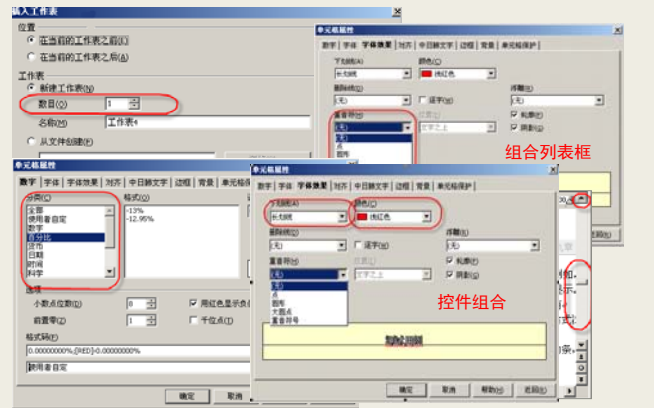
■ 复选框测试

- 多个复选框可以同时选中
- 多个复选框可以部分选中
- 多个复选框可以不被选中
- 逐一执行每个复选框的功能
- 组合执行复选框的功能



黑盒测试的方法

■ 常用控件测试



黑盒测试方法选择的综合策略

- 对于业务流程清晰的被测系统，可以利用场景法贯穿整个测试案例过程，对主要业务流程进行测试。
- 当主要流程测完后，再对系统中的重要功能进行等价类划分测试，将无限测试变成有限测试，这是减少工作量和提高测试效率最有效的办法。
- 在任何情况下都必须使用边界值分析法。经验表明，用这种方法设计的测试用例发现程序错误的能力最强。
- 可以依靠测试工程师的智慧和经验追加一些测试用例。
- 对照程序逻辑，检查已设计的测试用例的逻辑覆盖程度。如果没有达到要求的覆盖标准，应当再补充足够的测试用例。
- 对于参数配置类的软件功能，要用正交试验法选择较少的组合方式达到最佳效果。