

黑匣子测试

我们今天要讨论的内容

- ✓ 黑盒测试与白盒测试

- ✓ 黑匣子测试

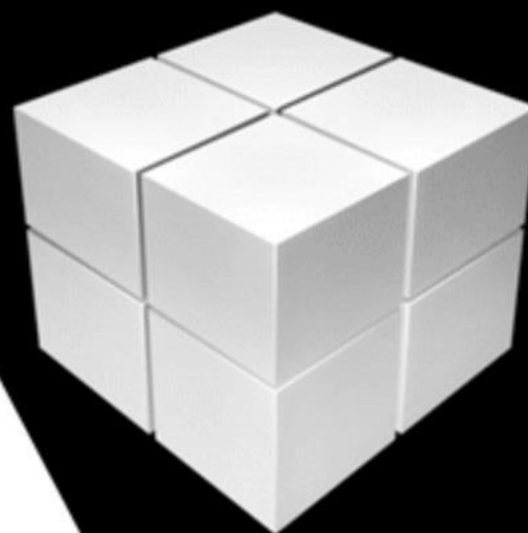
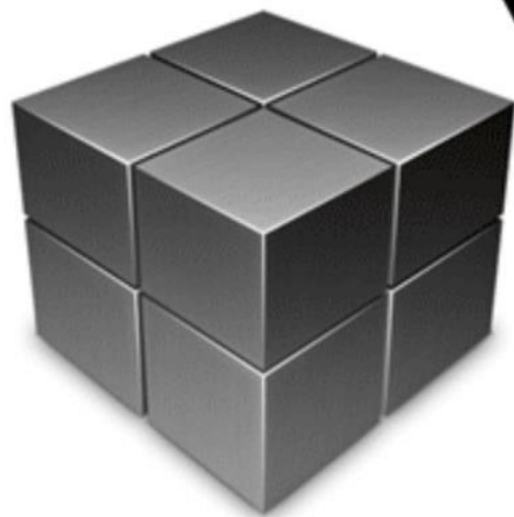
 - 等价分割

 - 边界值分析



黑盒测试与白盒测试

Black Box Vs White Box Testing



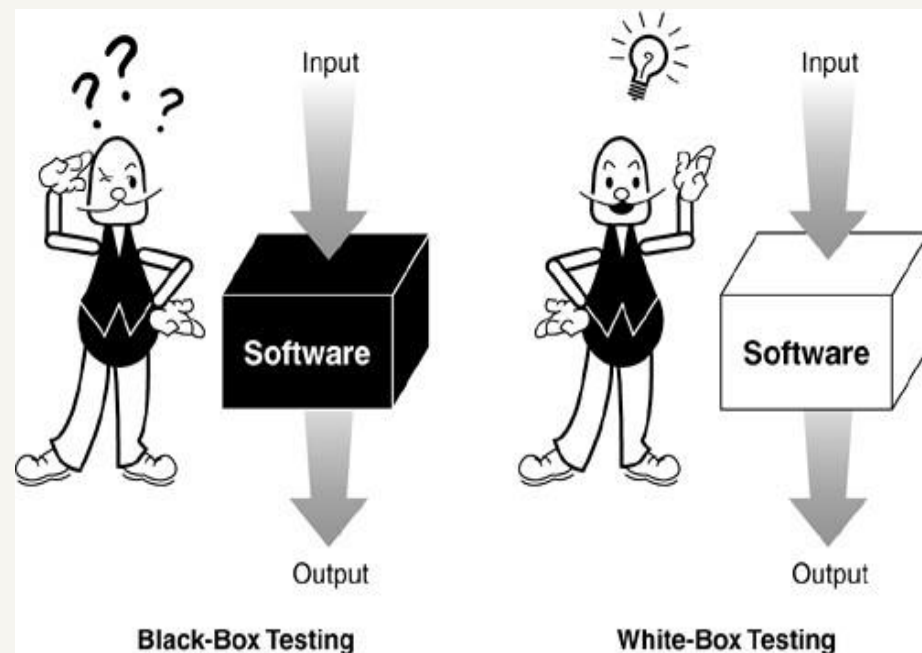
黑盒测试与白盒测试

✓ 黑匣子测试

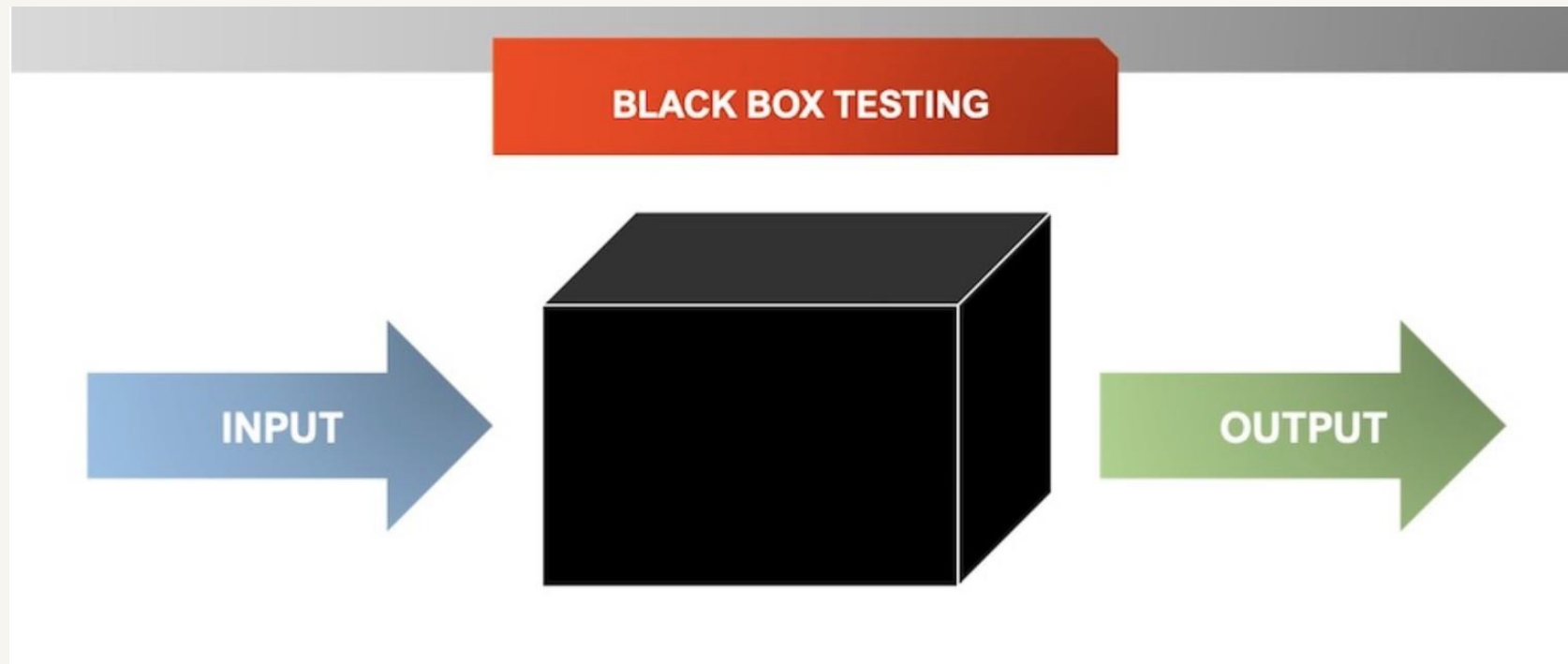
黑匣子测试是一种软件测试方法，测试人员不知道被测试项目的内部结构/设计/实现。

✓ 白盒测试

白盒测试是一种软件测试方法，其中被测试项目的内部结构/设计/实现是测试人员已知的。

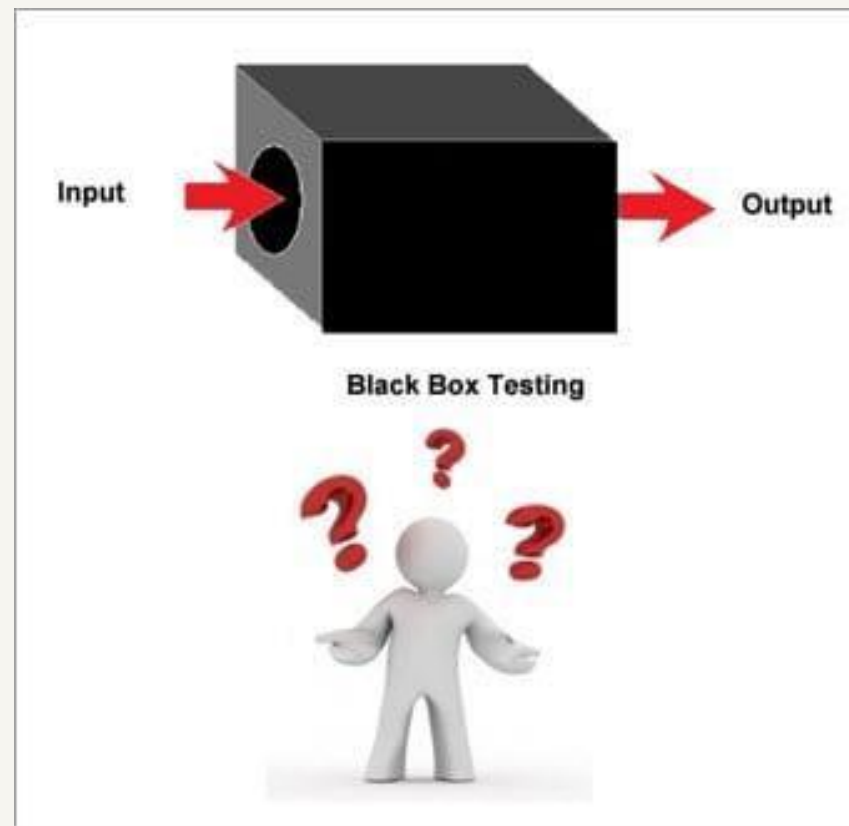


黑匣子测试



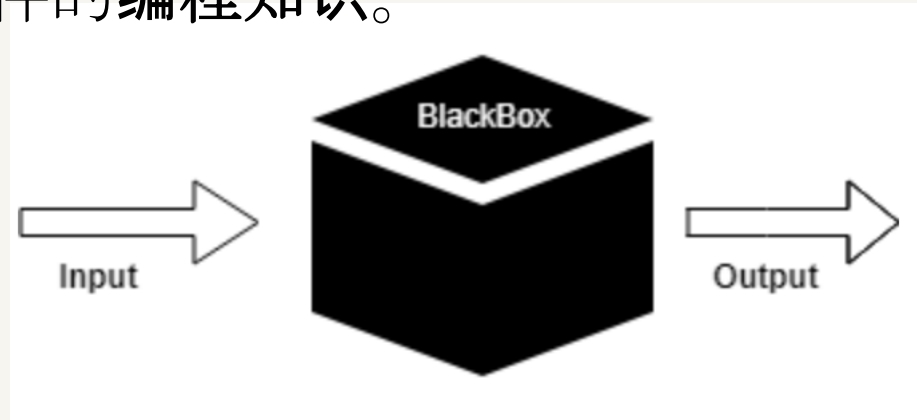
什么是黑匣子测试

- ✓ 黑盒测试检查软件的功能，而不窥视其内部结构或编码。
- ✓ 黑盒测试的主要来源是由客户提出的需求规格。



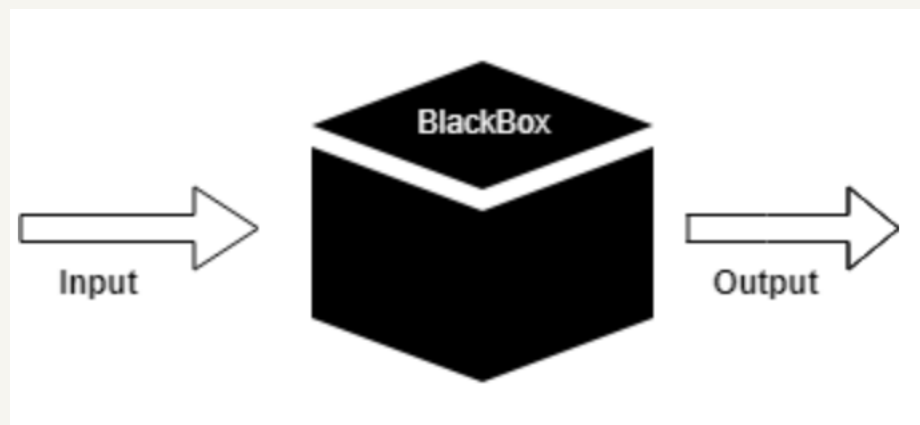
什么是黑匣子测试

- ✓ 测试者选择一个函数并给出**输入值**，以检查其功能，并检查该函数是否给出了**预期的输出**。
- ✓ 在完成所有功能的测试后，如果有严重的问题，就会把它交还给开发团队进行修正。
- ✓ 黑盒测试**不需要**软件的**编程知识**。



Question:

一旦你知道你要测试的输入和输出，你
认为下一步是什么？

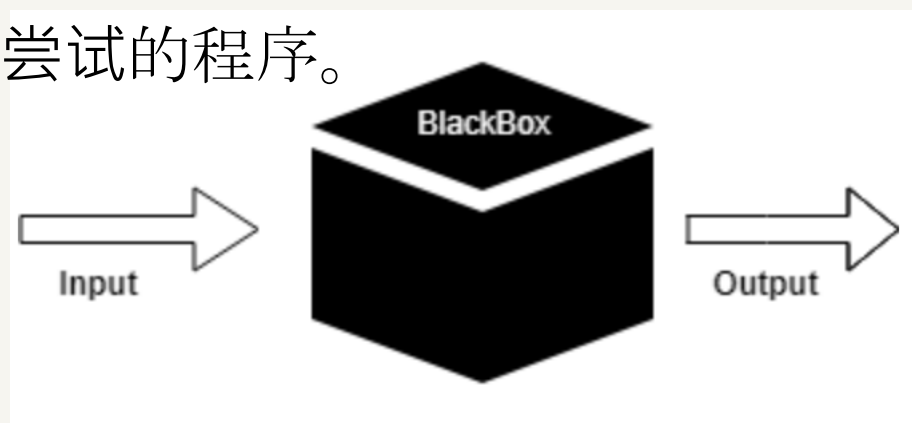


✓ 在黑盒测试中，你不需要被告知软件“盒子”里发生了什么。你只需要知道输入A会输出B，或者执行操作C会产生D。一个好的产品规格会给你提供这些细节。

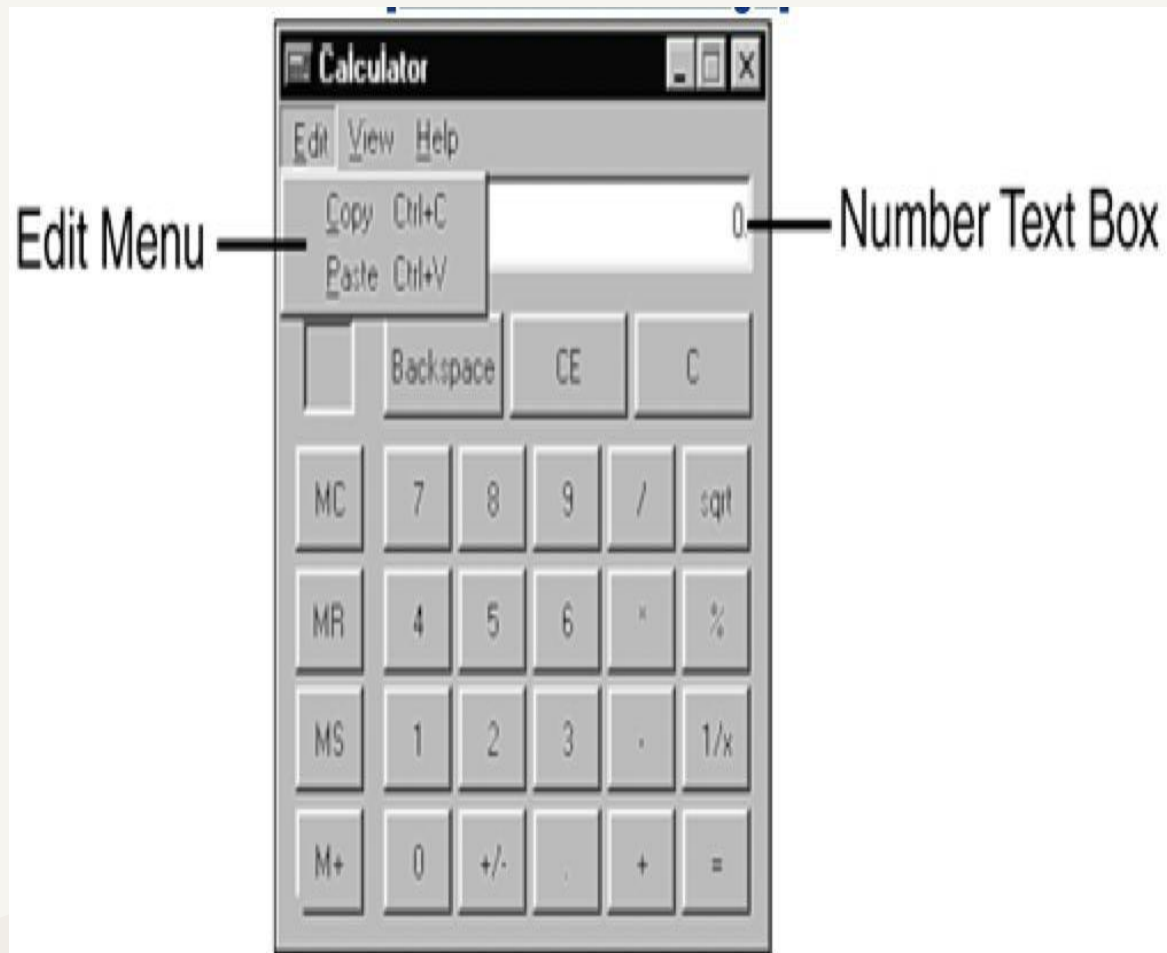
✓ 一旦你知道你要测试的软件的输入和输出，你的下一步就是开始定义测试案例。

✓ 测试用例是你将尝试的具体输入和你将尝试的程序。

当你测试该软件时，请遵循。



✓ 下图是你可能用来测试Windows计算器加法功能的几种情况的例子。



Addition Test Cases for Windows Calculator

| | | |
|--------|--------------|------|
| 0+0 | should equal | 0 |
| 0+1 | should equal | 1 |
| 254+1 | should equal | 255 |
| 255+1 | should equal | 256 |
| 256+1 | should equal | 257 |
| 1022+1 | should equal | 1023 |
| 1023+1 | should equal | 1024 |
| 1024+1 | should equal | 1025 |
| ... | | |
| ... | | |

一个计算器，加法功能，输入1-5。有多少种情况？

- ☐ A 1
- ☐ B 5
- ☐ C 25
- ☐ D 数不清的

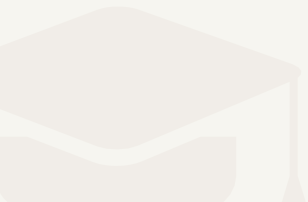
提交



✓ 为了用黑盒测试找到程序中的**所有bug**，你必须设计涵盖**所有可能的输入和输出条件的**测试用例。

✓ 但这是**不可能的**，因为耗尽测试的测试案例数量太大。

✓ **选择测试用例**是软件测试人员所做的**最重要的一项任务**。不正确的选择会导致测试太多，测试太少，或测试错误的东西。



黑匣子测试技术

✓ 有各种**测试用例设计技术**适用于黑盒测试。

- 等价分割

- 边界值分析

- 决策表测试

- 因果技术

等价分割



Equivalence Partitioning



什么是等价分割

- ✓ 等价划分，也被称为等价类划分 (ECP)。
- ✓ 在这种方法中，输入的领域数据被划分为不同的等价数据类别--一般被称为 "有效 "和 "无效"。
- ✓ 所有位于一个等价类中的数据项都被假定为在当作为输入时，被测试的软件应用程序以同样的方式传递。
- ✓ 等价分割的原则是，测试用例应设计为每个分区至少要覆盖一次。

什么是等价分割

✓ 等价分割是在某些假设下进行的。

□ 系统将在一个分区内处理所有的测试输入变化。

同样的方式。


□ 如果其中一个输入条件通过，那么该分区内的所有其他输入条件也将通过。

□ 如果其中一个输入条件失败，那么该分区内的所有其他输入条件也会失败

。

✓ 等价分割的成功和有效性在于上述假设的合理性。始终创建具有最大测试覆盖率

的最小测试案例。

- 
- ✓ 在等价分割中，我们只需要测试每个分区的一个条件。
 - ✓ 这是因为我们假设一个分区的所有条件都会被软件以同样的方式处理。
 - ✓ 如果一个分区中的一个条件起作用，我们假定该分区中的所有条件都会工作，因此测试这些其他的东西也没有什么意义。
 - ✓ 同样，如果一个分区中的一个条件不起作用，那么我们就假定该分区中的所有条件都不起作用，所以在该分区中再进行测试也没有什么意义。
 - ✓ 等价分割将测试用例的数量减少到一个涵盖最大可能性的可测试用例的有限列表。换句话说，我们可以保持测试覆盖率，同时我们可以减少返工量，最重要的是减少时间花了。

什么是等价分割

Equivalence Class Partitioning (ECP)

AGE * Accepts value from 18 to 60

| Equivalence Class Partitioning | | |
|--------------------------------|-------|-----------|
| Invalid | Valid | Invalid |
| ≤ 17 | 18-60 | ≥ 61 |

- ✓ 例如，在上图中，**AGE** 文本字段只接受从**18**到**60**的数字。
- ✓ 60. 将有三组班级或小组。
 - a) 小于或等于17。
 - b) 大于或等于61。
- ✓ 一个**有效的**班级将是18至60岁之间的任何人。
- ✓ 因此，我们将测试用例减少到只有3个基于形成的类的测试用例，从而涵盖了所有的可能性。因此，用每组类中的任何一个值进行测试，都足以测试上述情况。

什么是等价分割

MOBILE NUMBER

Enter Mobile No.

*Must be 10 digits

✓ 假设我们要测试一个接受十位数的手机号码的文件。

✓ 有效输入。10位数

✓ 无效输入。9位数，11位

✓ 有效等级。输入10位数的手机号码 = 9876543210

✓ 无效等级 输入少于10位的手机号码 = 987654321

✓ 无效等级 输入超过11位的手机号码 = 98765432109

| EQUIVALENCE PARTITIONING | | |
|--------------------------|------------|-------------|
| Invalid | Valid | Invalid |
| 987654321 | 9876543210 | 98765432109 |

什么是等价分割

✓ 等价分割的成功取决于我们是否有能力**创建**

正确的分区。

✓ 然而，你应该知道，我们是**作为一个黑匣子**来测试这个应用程序的。因此，

我们创建分区的能力仅限于需求中所呼吁的内容。我们**不了解**设计，也不知道开发人员会编什么代码。

什么是等价分割

- ✓ 假设一个应用程序接受一个数字作为输入，其值在10到100之间。现在，使用等价类划分，创建可能的等价类。

| Equivalence Class | Explanation |
|-------------------|--|
| Numbers 10 to 100 | This class will include test data for a positive scenario. |
| Numbers 0 to 9 | This class will include test data that is restricted by the application. Since it is designed to work with numbers 10 to 100 only. |
| Greater than 100 | This class will again include test data that is restricted by the application but this time to test the upper limit. |

你认为还有没有其他的等价类？

- ✓ 覆盖所有测试数据类型的正反面测试数据。我们必须以这样的方式创建测试数据类，以**涵盖最大的测试覆盖率**，但同时，**不应该有任何形式的冗余**。

| Equivalence Class | Explanation |
|-------------------|---|
| Numbers 10 to 100 | This class will include test data for a positive scenario. |
| Numbers 0 to 9 | This class will include test data that is restricted by the application. Since it is designed to work with numbers 10 to 100 only. |
| Greater than 100 | This class will again include test data that is restricted by the application but this time to test the upper limit. |
| Negative numbers | Since negative numbers can be treated in a different way so, we will create a different class for negative numbers in order to check the robustness of the application. |

- ✓ 该应用程序不能与小于**10**的数字。
- ✓ 因此，我们没有为小于**10**的数字创建**1**个类，而是创建了两个类--数字**0-9**和负数。
- ✓ 这是因为，应用程序有可能以不同的方式处理负数。

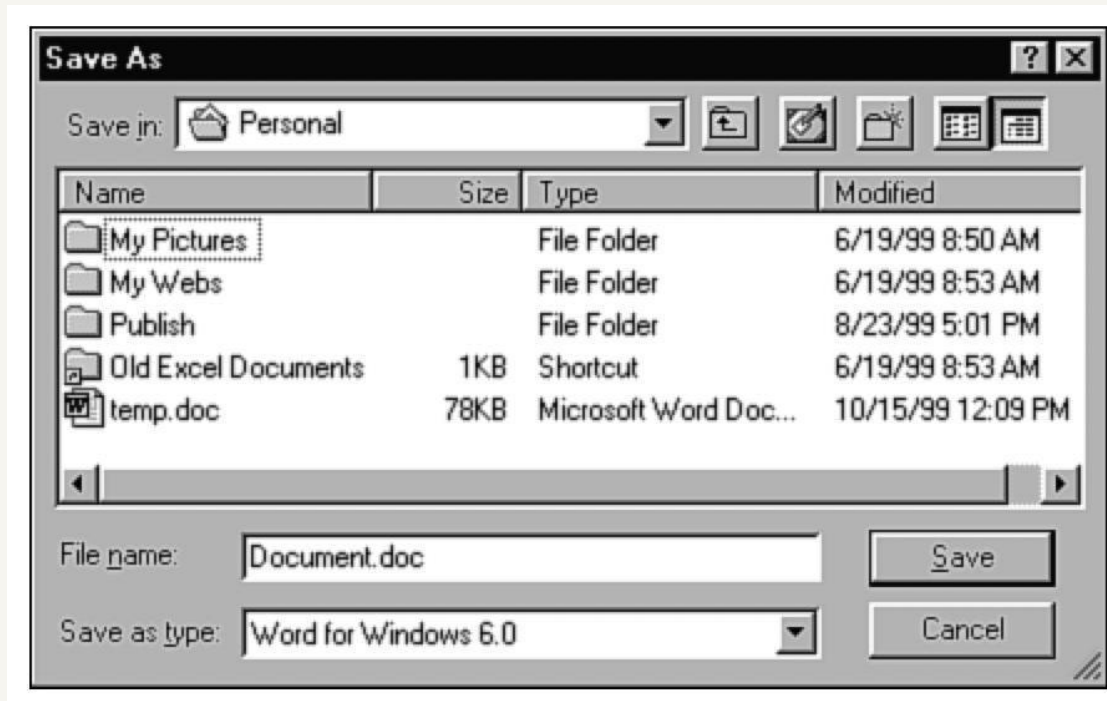
✓ 如果有可能对某一特定类别的测试数据进行不同的处理，那么最好将该

等价类拆开。

| Equivalence Class | Explanation |
|--------------------|---|
| Numbers 10 to 100 | This class will include test data for a positive scenario. |
| Numbers 0 to 9 | This class will include test data that is restricted by the application. Since it is designed to work with numbers 10 to 100 only. |
| Greater than 100 | This class will again include test data that is restricted by the application but this time to test the upper limit. |
| Negative numbers | Since negative numbers can be treated in a different way so, we will create a different class for negative numbers in order to check the robustness of the application. |
| Alphabets | This class will be used to test the robustness of the application with non-numeric characters. |
| Special characters | Just like the equivalence class for alphabets, we can have a separate equivalence class for special characters. |

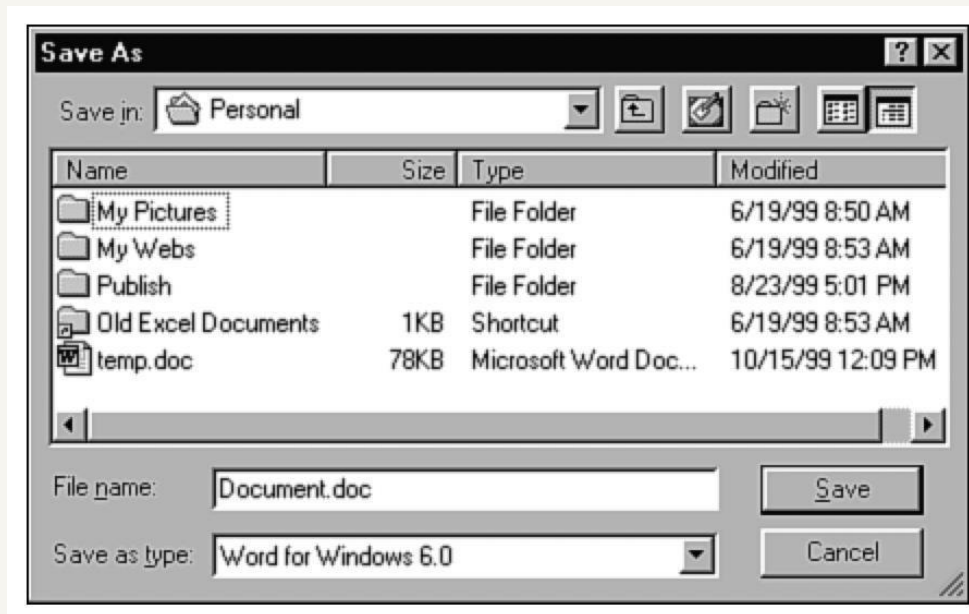
Question:

- ✓ 一个Windows文件名可以包含任何字符，除了 \ : * ? " < > 和 |。文件名可以有1到255个字符。



- ✓ 假设你正在创建文件名的测试案例，列出可能的等价分区。

- ✓ 一个Windows文件名可以包含任何字符，除了 \ / : * ? " < > 和 |。文件名可以有1到255个字符。



- 1:具有有效字符的文件名
- 2:具有无效字符的文件名
- 3:具有有效长度的文件名
- 4:文件名长度小于1个字符，无效。
- 5：文件名的长度无效，超过255个字符

创建等价分割类的规则

✓ 采用以下准则来协助定义等价类。

如果一个输入条件指定了一个范围，则定义了一个有效和两个无效的等价类：

在范围内是有效的，过大或过小是无效的。

例子。年龄是一个从1-99的数字。

□ 有效的等价类。[1 - 99]

□ 无效的等价类。[少于1]

□ 无效的等价类。[大于99]

创建等价分割类的规则

✓ 采用以下准则来协助定义等价类。

如果一个输入条件需要一个枚举集的成员，而所有的值都是一样的，那么就为该集的值定义了一个有效的类和一个无效的等价类。

例子。San Luis Obispo = {SLO, San Luis, Slo Town}。

- ❑ 有效的等价物类。 [SLO, San Luis, Slo镇]
- ❑ 无效的等价类。 [庇斯莫]。

创建等价分割类的规则

✓ 采用以下准则来协助定义等价类。

如果一个输入条件需要一个**列举集的成员**，而每个**列举集**都被**区别对待**，那么就为列举集的每个值定义一个有效的类和一个无效的等价类。

例子。车辆={汽车、卡车、摩托车}。

□ 有效的等价类。 [汽车]

□ 有效的等价类。 [卡车]。

□ 有效的等价物类别。 [摩托车]。

□ 无效的等价物类。 [滑板]

创建等价分割类的规则

✓ 采用以下准则来协助定义等价类。

如果一个输入条件是**布尔值**，则定义了两个有效的类。

例如：要求隔夜运输

□ 有效的等价类。 [是]。

□ 有效的等价类。 [无]

创建等价分割类的规则

✓ 采用以下准则来协助定义等价类。

问题域定义了一些特征，这些特征区分了一些来自他人的投入。

例如：正方形是一个四边形的多边形，所有的边都是等长的。

□ 有效的等价类。 [5,5,5,5]

□ 无效的等价类。 [5,5,6,6]

等值分割的优势

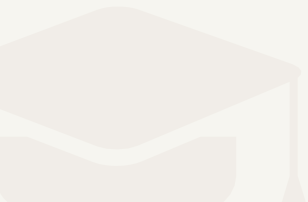
- ✓ 在等价类测试的帮助下，测试用例的数量可以是大大减少，同时保持相同的测试覆盖率。
- ✓ 这种测试技术有助于在最短的时间内交付高质量的产品。
- ✓ 它完全适用于有时间和资源限制的软件项目。



等价分割的缺点

- ✓ 等价类测试的整个成功依赖于**对等价类的识别**。这些类的识别依赖于**测试人员**创建这些类和基于它们的测试用例的**能力**。
- ✓ 在**复杂**应用的情况下，要确定**所有**的一组数据是非常**困难**的。

等价类，需要测试人员方面有大量的专业知识。
- ✓ **不正确地**识别等价类会导致**较小的测试覆盖率**和**缺陷泄漏的可能性**。



Question:

你能从下面的例子中发现任何问题吗？

Equivalence Class Partitioning (ECP)

AGE

Enter Age

* Accepts value from 18 to 60

| Equivalence Class Partitioning | | |
|--------------------------------|-------|-----------|
| Invalid | Valid | Invalid |
| ≤ 17 | 18-60 | ≥ 61 |

靠近边界的输入值出错的几率更高。

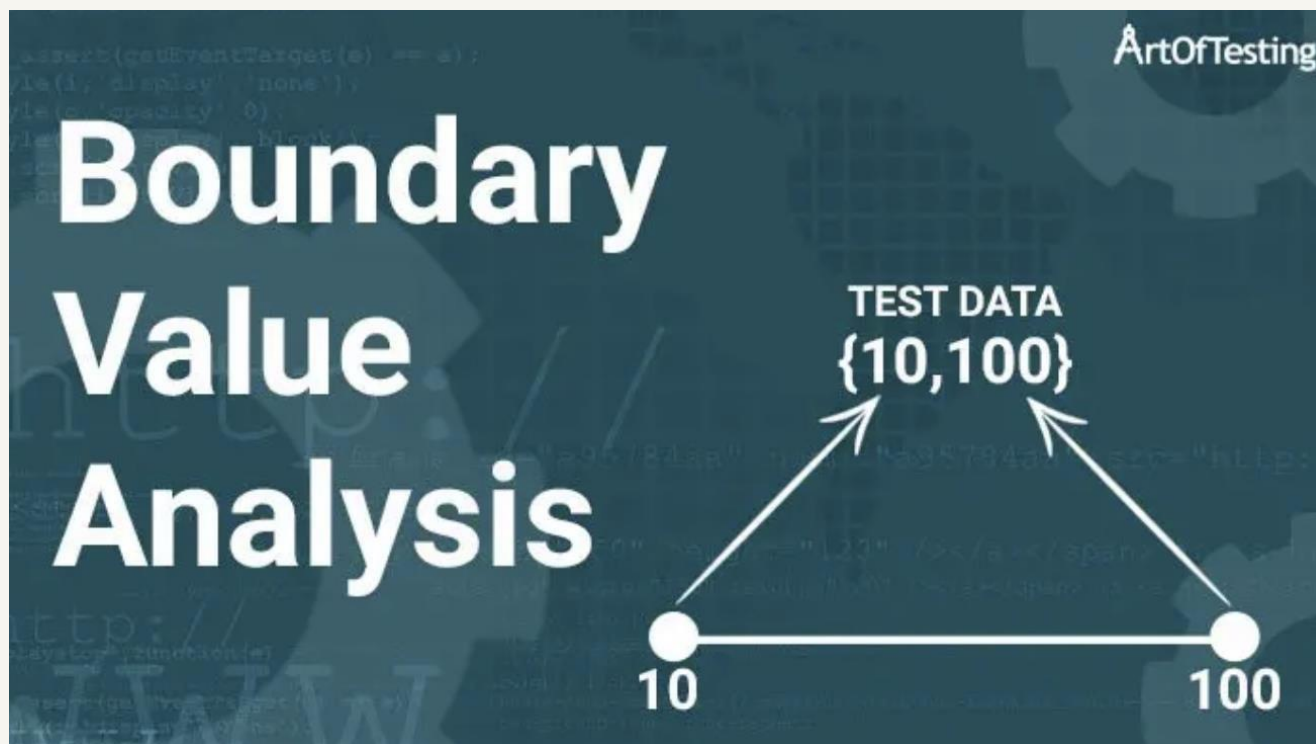
Y2K (2000年) 问题：边界的错误

- 2000年问题：许多程序只用最后两位数字表示四位数的年份，使2000年与1900年无法区分。





边界值分析



什么是边界值分析

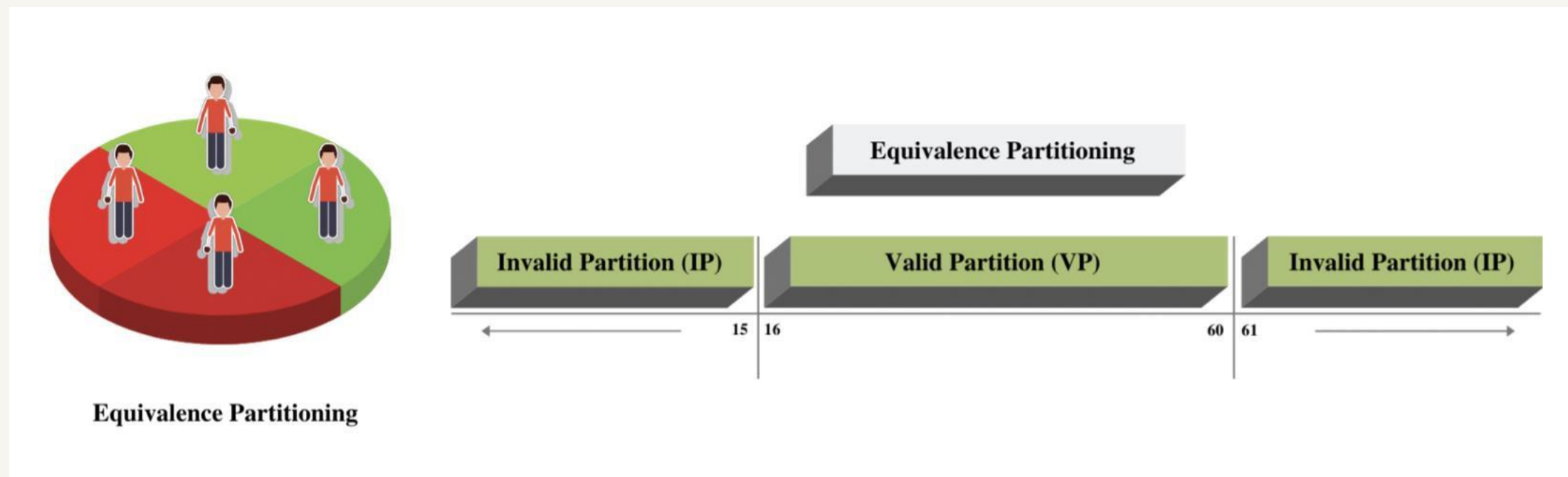
- ✓ 更多的应用**错误**发生在输入域的**边界处**。
- ✓ 边界值分析是用来**识别边界处的错误**，而不是用来识别找到那些存在于输入域中心的。
- ✓ 边界值分析是等价划分的下一个部分，用于设计测试用例，测试用例被选在等价类的**边缘**。

什么是边界值分析

- ✓ 边界值分析是基于**测试有效和无效分区**的**边界值**。
- ✓ 每个分区都有它的**最大值**和**最小值**，这些最大值和最小值是一个分区的**边界值**。
- ✓ 测试可以被设计为**涵盖有效和无效的边界值**。
- ✓ 在设计测试用例时，对**每个边界值都要选择一个测试**。同时，对于每个边界，
我们测试边界两边的最小有效数字的**+/-1**。

如何进行边界值分析

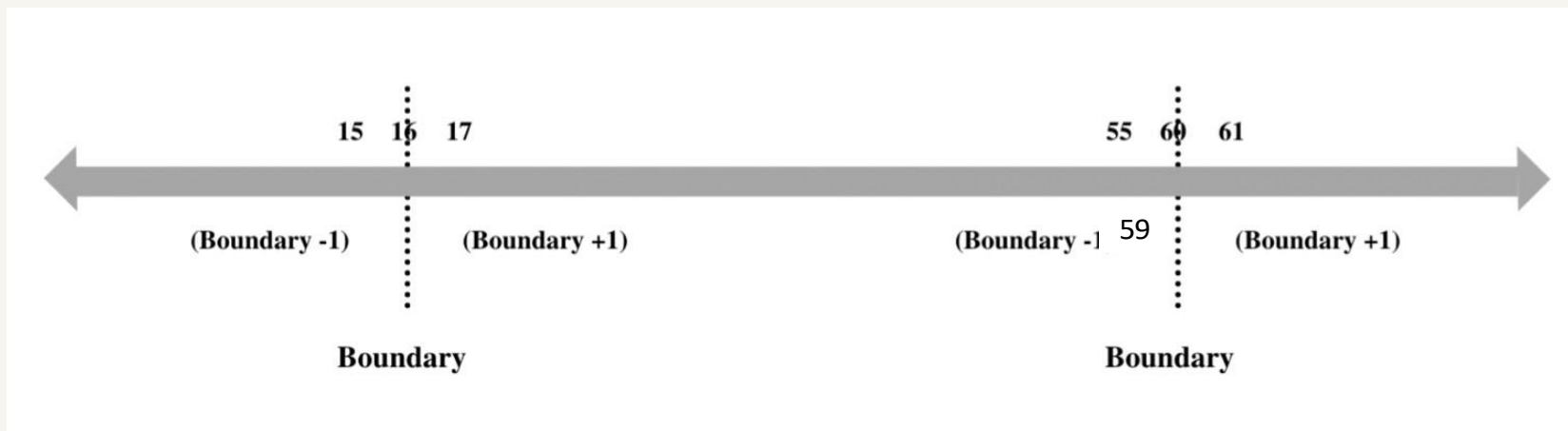
假设一个健身房只允许年龄在**16到60岁**之间的人（ **$16 \leq \text{年龄} \leq 60$** ）进行登记。



✓ 边界值分析的第一步是建立等值分割。

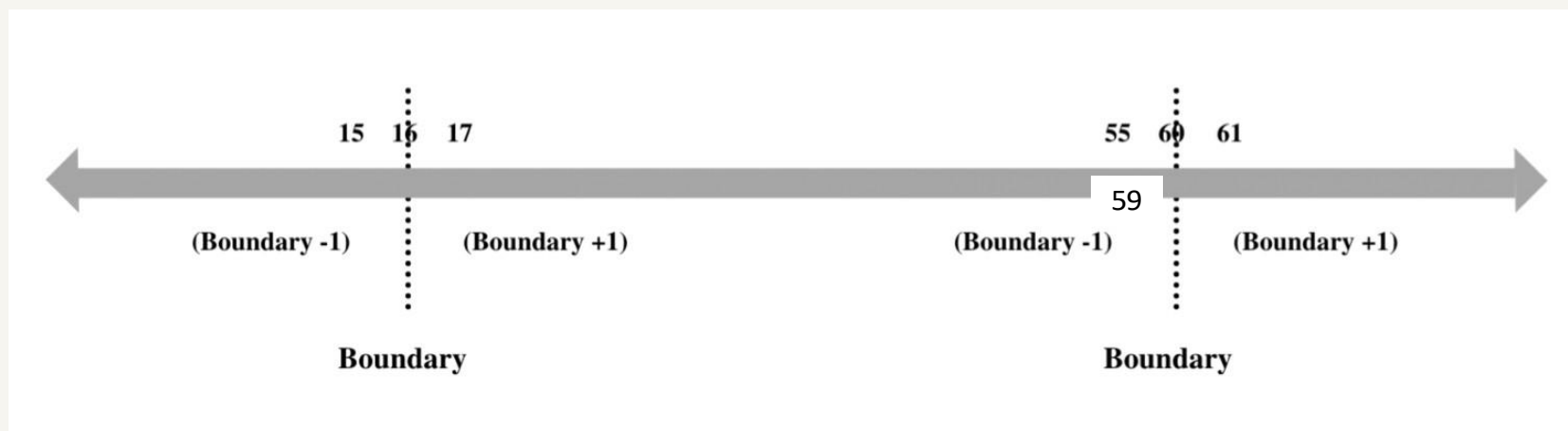
如何进行边界值分析

我们有一个3步方法来确定边界。



- ✓ 确定**这个分区类的精确边界值**--是16和60。
- ✓ 获得比准确的边界值少一个的边界值--也就是15和59。
- ✓ 获得比精确边界多一个的**边界值**--也就是17和20。
61 .
- ✓ 如果我们把它们全部结合起来，我们将得到以下年龄标准的边界值组合。
 - 有效的边界条件:年龄=16, 17, 59, 60
 - 无效的边界条件:年龄=15, 61。

如何进行边界值分析



测试案例1：输入数值15 ($16-1$) = 无效

测试案例2：输入数值16=有效

测试用例3：输入数值17 ($16+1$) =有效

测试用例4：输入数值59 ($60-1$) =有效 测

试用例5：输入数值60=有效

测试用例6:输入值61 ($60+1$) =无效

什么是边界值分析

- ✓ 假设我们要测试一个文本字段（姓名），它的长度在**6-12**个字符之间。

Name *Accepts characters length (6 - 12)

最小边界值为6

最大边界值为12有效文本长

度为6、7、11、12

无效文本长度为5、13

| BOUNDARY VALUE ANALYSIS | | |
|-------------------------|---------------------------------|---------------------|
| Invalid (min -1) | Valid (min, +min, -max, max) | Invalid (max +1) |
| 5 characters | 6, 7, 11, 12 characters | 13 characters |

什么是边界值分析

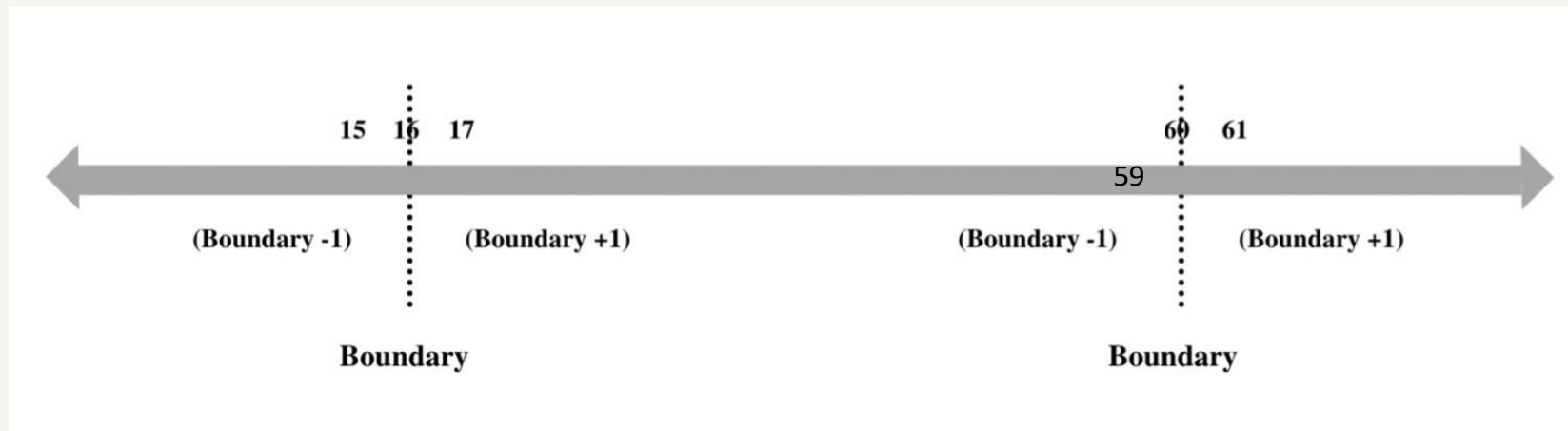
✓ 假设我们要测试一个文本字段（姓名），它的长度在6-12个字符之间。

| Name <input type="text" value="Enter Name"/> *Accepts characters length (6 - 12) | | |
|--|---------------------------------|---------------------|
| BOUNDARY VALUE ANALYSIS | | |
| Invalid (min -1) | Valid (min, +min, -max, max) | Invalid (max +1) |
| 5 characters | 6, 7, 11, 12 characters | 13 characters |

测试案例1：文本长度为5（min-1）= 无效 测试用例2：文本长度正好为6（最小值）=有效
测试用例3：文本长度为7（最小值+1）=有效 测试案例4：文本长度为11（最大1）=有效
测试案例5：文本长度正好为12（最大）=有效
测试案例6:文本长度为13（最大+1）= 无效

Question:

对于每个边界，我们为什么要测试边界两边的最小有效数字的 ± 1 ？



假设一个健身房只允许年龄在 **$16 < \text{年龄} \leq 60$** 之间的人注册。



✓ cLheant~~one~~已经写了下面的逻辑。

开发商

- 如果（年龄 ≤ 17 岁），则不允许加入健身房。
- 如果（年龄 > 60 岁），则不允许加入健身房。
- 如果你看一下逻辑，你会发现，逻辑应该是If (*age* < 17)，但开发者添加了=错误的符号。

等价分割的区别 和边界值分析

| Boundary value analysis | Equivalence partitioning |
|--|---|
| It is a technique where we identify the errors at the boundaries of input data to discover those errors in the input center. | It is a technique where the input data is divided into partitions of valid and invalid values. |
| Boundary values are those that contain the upper and lower limit of a variable. | In this, the inputs to the software or the application are separated into groups expected to show similar behavior. |
| Boundary value analysis is testing the boundaries between partitions. | It allows us to divide a set of test conditions into a partition that should be considered the same. |

Assignment



- ✓ 假设你要测试一个软件。
- ✓ 输入是一个三角形的三条边的长度，分别表示为a、b、c。
- ✓ 当每条边都是小于或等于**20**的正数且三角形是等腰三角形时，输出为真，
否则输出为假。
- ✓ 使用ECP和边界值分析为软件设计测试案例

等价分割的缺点 和边界值分析

一个方案。

输入：2000年至2004年2月的8位数字的日期。

如：20010206

输出：下一个日期

如：20010207

等价类和界线清楚吗？

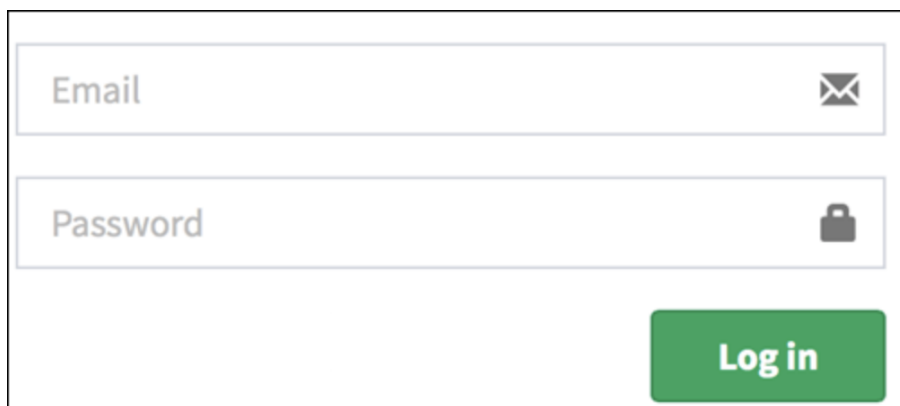
等价分割和边界值分析假定输入变量

是独立的。



决策表测试

让我们为一个登录屏幕设计测试用例。



The image shows a login form with two input fields. The first field is labeled 'Email' and has an envelope icon on the right. The second field is labeled 'Password' and has a lock icon on the right. Below these fields is a green button with the text 'Log in'.

问题。你能用等价分割法或边界值分析法为其设计测试用例吗？

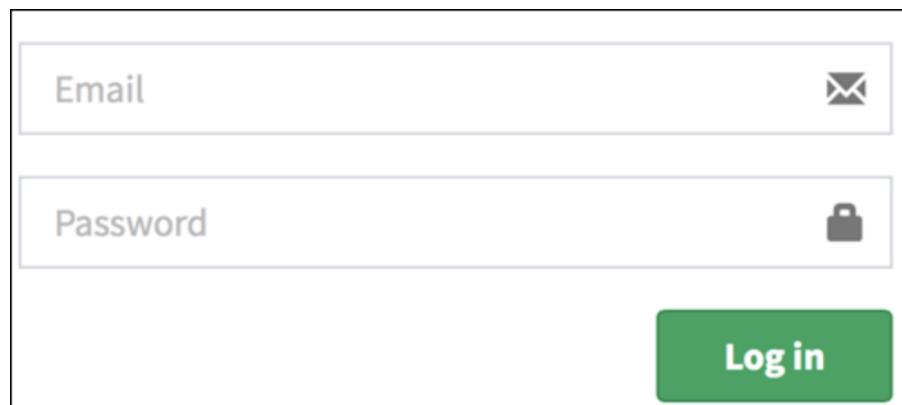
- ✓ 对于登录屏幕，如果用户提供了正确的用户名和密码，用户将被重定向到主页。如果有任何输入错误，将显示一个错误信息。

- ✓ 等价分割和边界值分析只适用于特定条件或输入。
- ✓ 决策表测试用于测试不同输入组合的系统行为。
- ✓ 在决策表中，不同的输入组合及其相应的系统行为（输出）都以表格的形式记录下来。
- ✓ 决策表的每一列代表一个测试案例。

例1：如何为登录屏幕制作决策表

52

让我们为一个登录屏幕创建一个决策表。



The image shows a login form with two input fields. The first field is labeled 'Email' and has an envelope icon on the right. The second field is labeled 'Password' and has a lock icon on the right. Below these fields is a green button with the text 'Log in'.

- ✓ 条件很简单，如果用户提供了正确的用户名和密码，用户就会被重定向到主页上。

✓ 如果任何一个输入是错误的，将显示一个错误信息。

例1：如何为登录屏幕制作决策表

让我们为一个登录屏幕创建一个决策表。

| Conditions | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
|----------------|--------|--------|--------|--------|
| Username (T/F) | F | T | F | T |
| Password (T/F) | F | F | T | T |
| Output (E/H) | E | E | E | H |

T - 正确的用户名/密码

F - 错误的用户名/密码 E - 显

示错误信息

H - 显示主屏幕

例1：如何为登录屏幕制作决策表

54

| Conditions | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
|----------------|--------|--------|--------|--------|
| Username (T/F) | F | T | F | T |
| Password (T/F) | F | F | T | T |
| Output (E/H) | E | E | E | H |

- ❑ 案例1 - 用户名和密码都是**错误的**。用户会看到一个**错误**信息。
- ❑ 案例2 - 用户名是**正确的**，但密码是**错误的**。用户会看到一个**错误**信息。
- ❑ 案例3 - 用户名是**错误的**，但密码是**正确的**。该用户显示一个**错误**信息。
- ❑ 案例4 - 用户名和密码都是**正确的**，并且用户浏览到了主页

什么是决策表

55

| | | rules | | | | | | | | | |
|------------|----|-------|----|----|----|----|----|----|----|----------------------|--|
| | | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | | |
| conditions | C1 | T | T | T | T | F | F | F | F | values of conditions | |
| | C2 | T | T | F | F | T | T | F | F | | |
| | C3 | T | F | T | F | T | F | T | F | | |
| actions | a1 | x | | | x | x | | | x | actions taken | |
| | a2 | x | | | | | | | x | | |
| | a3 | | x | | | | x | | | | |
| | a4 | | | x | x | | | x | x | | |
| | a5 | x | | | x | | | | | | |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第1步：分析需求并创建第一列

- 要求。"如果要求的金额被余额所覆盖，或者如果客户被授予信贷以覆盖提款金额，则允许提款"。
- 在一个列表中表达条件和结果的行动，使它们要么为真，要么为假。在这个例子中，有两个条件，"取款金额 \leq 余额"和"授予信贷"。有一个结果，提款被批准。

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第1步：分析需求并创建第一列

| |
|----------|
| 条件 |
| 提款金额<=余额 |
| 授信 |
| 行动 |
| 准予退学 |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第2步：添加列

- 计算表格中需要多少列。列的数量取决于条件的数量和每个条件的备选方案的数量。如果有两个条件，每个条件可以是真或假，你需要4列。如果有三个条件，就需要8列，以此类推。
- 在数学上，列的数量是2个条件。在这种情况下， $2^2=4$ 列。

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第2步：添加列

| 条件 | | | | |
|----------|---|---|---|---|
| 提款金额<=余额 | T | F | T | F |
| 授信 | T | T | F | F |
| 行动 | | | | |
| 准予退学 | | | | |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第3步：缩表

- 用"-"标记不重要的数值。如果要求的金额小于或等于账户余额，那么是否给予信贷并不重要。
- 在下一步中，你可以删除那些已经变得相同的列。

创建决策表的步骤

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第3步：缩表

| 条件 | | | | |
|----------|---|---|---|---|
| 提款金额<=余额 | T | F | T | F |
| 授信 | T | T | F | F |
| 行动 | | | | |
| 准予退学 | | | | |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户中有足够的资金，或者客户有足够的信用额度，那么他们就可以获得更多的钱

第3步：缩表

| 条件 | | | | |
|----------|---|---|---|---|
| 提款金额<=余额 | T | F | T | F |
| 授信 | - | T | - | F |
| 行动 | | | | |
| 准予退学 | | | | |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第3步：缩表

| 条件 | | | |
|----------|---|---|---|
| 提款金额<=余额 | T | F | F |
| 授信 | - | T | F |
| 行动 | | | |
| 准予退学 | | | |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第4步：确定行动

□ 为表格中的每一列输入行动。你将能够在要求中找到这些信息。

□ 命名各列（规则）。它们可以被命名为R1/规则1，R2/规则2等等，但你也可以给它们起更多描述性的名字。

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第4步：确定行动

| 条件 | R1 | R2 | R3 |
|----------|----|----|----|
| 提款金额<=余额 | T | F | F |
| 授信 | - | T | F |
| 行动 | | | |
| 准予退学 | T | T | F |

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户中有足够的资金，或者客户有足够的信用额度，那么他们就可以获得更多的钱

第5步：编写测试案例

- 根据该表编写测试用例。
- 每一栏至少有一个测试用例，可以全面覆盖所有的业务规则。

假设我们想测试ATM机的以下取款功能。

假设一个客户要求从ATM机上提取现金。ATM机支付了

如果客户的账户有足够的资金，或者客户有足够的信用额度，那么就可以将这笔钱转给客户。

第5步：编写测试案例

| 条件 | R1 | R2 | R3 |
|-------------|----|----|----|
| 提取金额 ≤平衡 | T | F | F |
| 授信 | - | T | F |
| 行动 | | | |
| 准予退学 | T | T | F |

- ①R1的 测试案例：余额=200，请求提款
= 200.预期结果：批准提款。
- ②R2的测试案例：余额=100，要求提款
=200，给予学分。预期结果：退学
准予。

③R3的 测试案例：余额

=100，请求提款

=200，没有学分。预期

结果：提款被拒。

例1：三角形问题

69

✓ 考虑一条程序语句，给定3条边的长度，确定这3条边是否能

(i) 形成一个三角形和

(ii) 哪种类型的三角形（等边、等腰、或斜边）。

| |
|---|
| C1: $\langle a, b, c \rangle$ forms a triangle? |
| C3: $a = b$? |
| C4: $a = c$? |
| C5: $b = c$? |
| A1: Not a Triangle |
| A2: Scalene |
| A3: Isosceles |
| A4: Equilateral |

问题：填表。

例1：三角形问题

70

✓ 考虑一条程序语句，给定3条边的长度，确定这3条边是否能

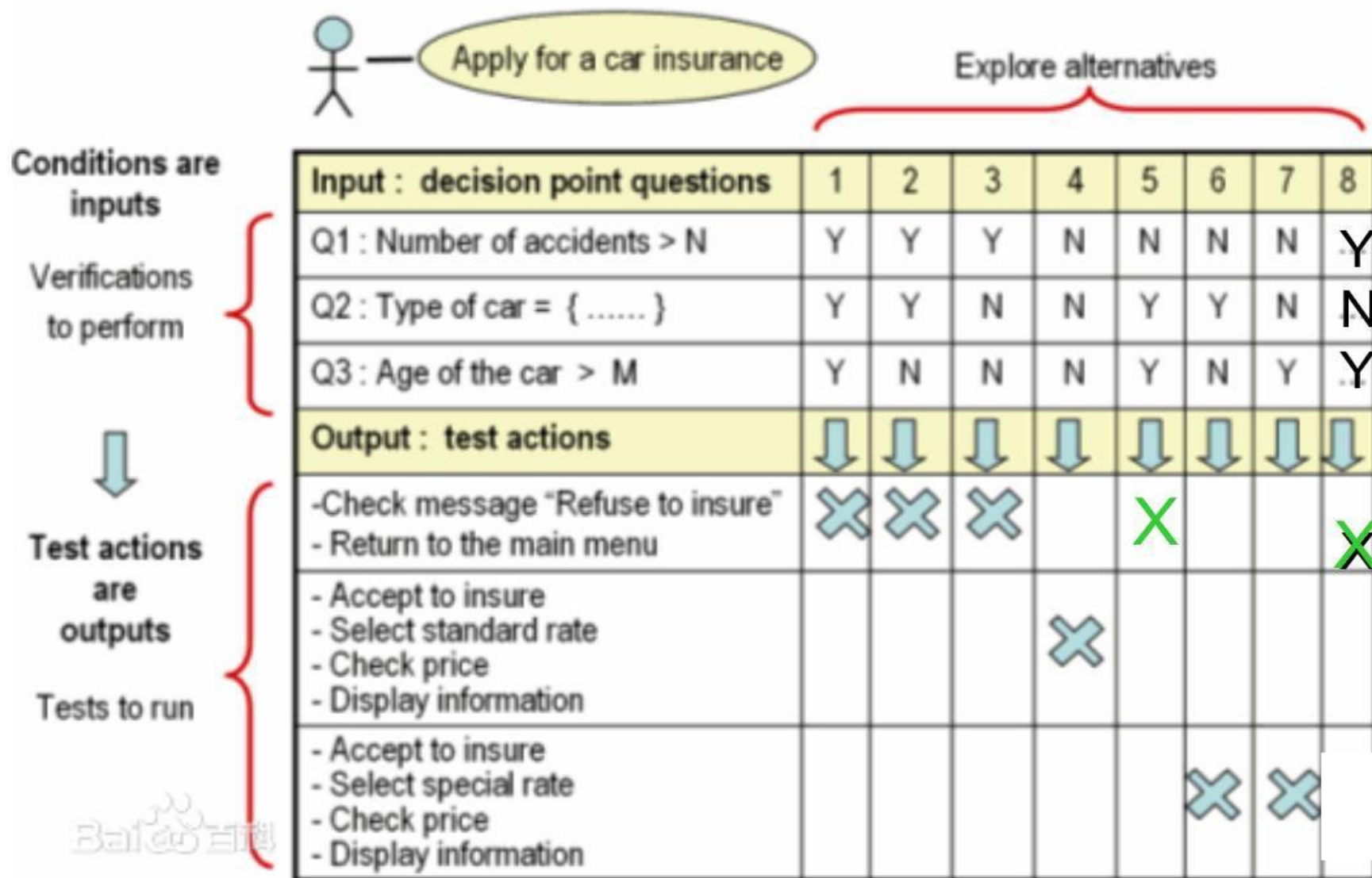
(i) 形成一个三角形和

(ii) 哪种类型的三角形（等边、等腰、或斜边）。

| | | | | | | | | | |
|---|---|---|----------|----------|---|----------|---|---|---|
| C1: $\langle a, b, c \rangle$ forms a triangle? | F | T | T | T | T | T | T | T | T |
| C3: $a = b$? | - | T | T | T | T | F | F | F | F |
| C4: $a = c$? | - | T | T | F | F | T | T | F | F |
| C5: $b = c$? | - | T | F | T | F | T | F | T | F |
| A1: Not a Triangle | X | | | | | | | | |
| A2: Scalene | | | | | | | | | X |
| A3: Isosceles | | | | | X | | X | X | |
| A4: Equilateral | | X | | | | | | | |
| A5: Impossible | | | X | X | | X | | | |

例2：汽车保险

71

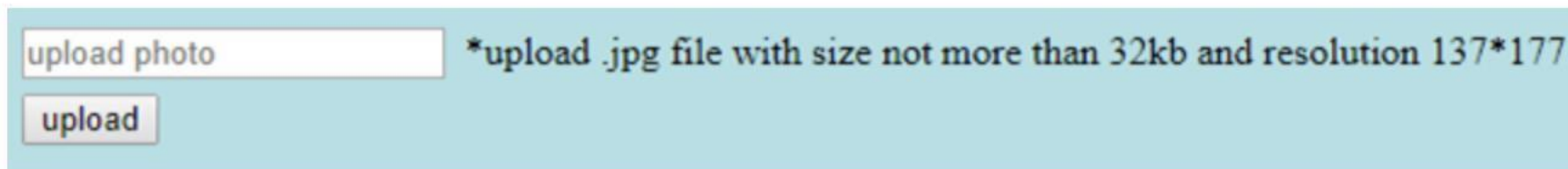


- ✓ 当系统行为对于不同的输入是不同的，而对于一系列的输入是不一样的，等效分区和边界值分析都没有帮助，但可以使用决策表。
- ✓ 这张表将有助于进行有效的组合，并能确保测试有更好的覆盖率。
- ✓ 在我们要追求100%覆盖率的情况下，通常当输入组合较少时，这种技术可以确保覆盖率。

- ✓ 主要的缺点是，当**输入的数量**增加时，表格将变得**更加复杂**。

现在考虑一个对话框，它将要求用户上传带有以下内容的照片某些条件。

- ❑ 你可以只上传'.jpg'格式的图片
- ❑ 文件大小小于32kb
- ❑ 分辨率必须是137*177。



The image shows a light blue rectangular dialog box. On the left, there is a white text input field with the placeholder text "upload photo". To the right of the input field, there is a line of text: "*upload .jpg file with size not more than 32kb and resolution 137*177". Below the input field, there is a small button with the text "upload".

- ❑ 如果任何一个条件失败，系统将抛出相应的**错误**

说明该问题的消息。

□ 如果所有条件得到满足，照片将被成功更新。