

## 样卷 1 参考答案

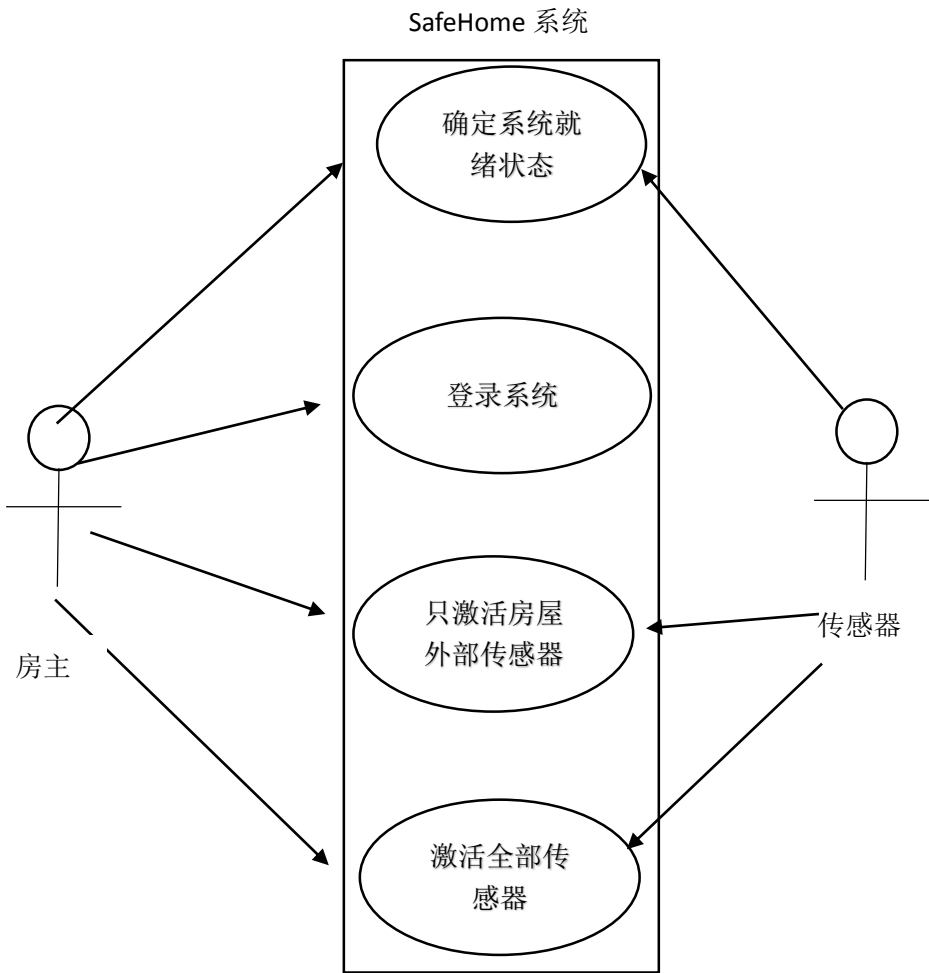
### 一、by 姚远

- 1.即使是很小的项目或者程序，在写代码时也要考虑到可复用性、可阅读性、易维护性。当有多人协作时就更是这样，要有良好的严谨的代码风格和接口定义，这样才能提高效率。
2. 添加程序员赶进度：实际上导致进度下滑：新人必须学习系统，在教学中，教的人没有完成工作，而且沟通渠道和固有延迟会增加。
- 3.这是正确的。信息隐藏是面向对象的程序设计方法的重要原则。这样可以大大降低模块之间的耦合性，使模块之间划分清晰。保证数据安全，使实现、测试、维护都更加容易。尤其适用于大型项目。
- 4.测试应该尽早介入工程开发过程。因为越早进行测试，发现问题和修正问题的成本越小。在项目后期进行测试，发现问题可能会导致系统架构发生重大变化，严重拖延进度。因此测试不应该在编码结束之后才进行。
- 5.正确运行只是高质量软件的一部分。高质量的软件还需要易于维护、易于复用、能够针对需求的变化容易的进行调整、有较好的可拓展性。

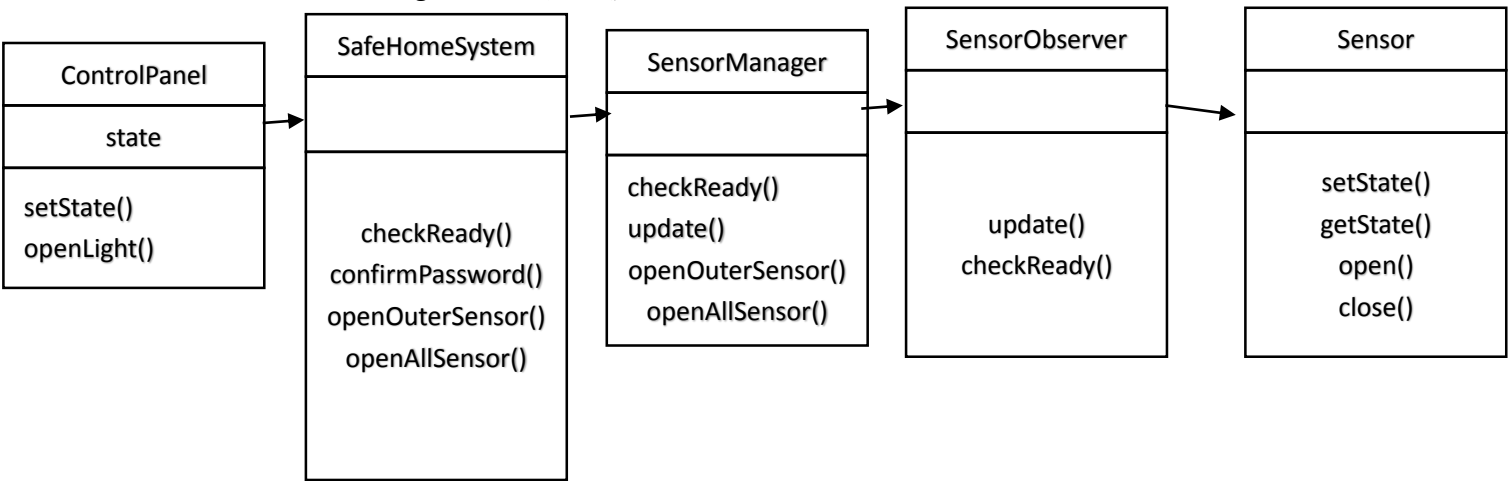
出处是 slide1 中的 software myths

二、by 王栋

1. 分析系统的主要功能， 采用 UML 用例图（ Use Case Diagram） 建立需求模型。

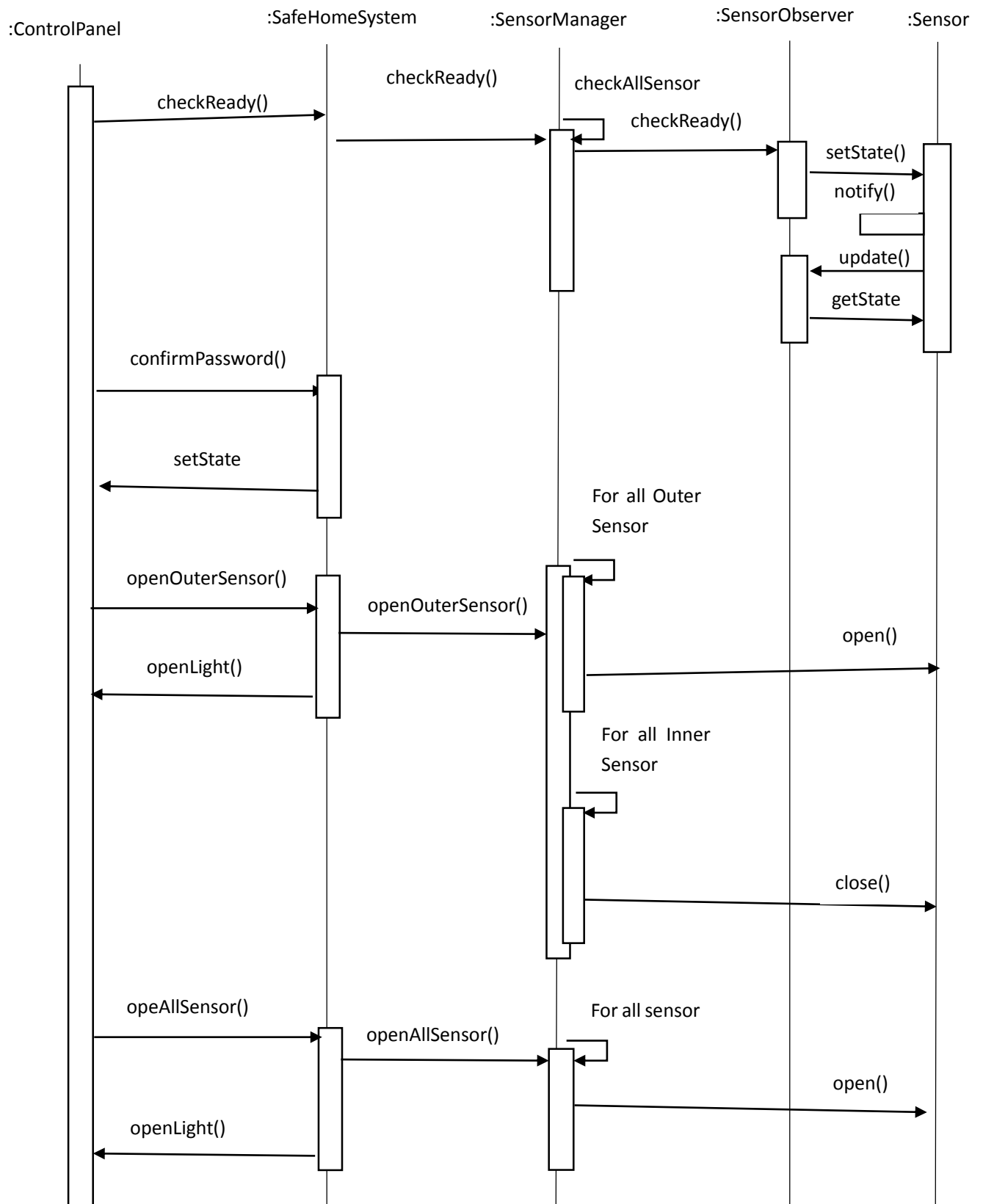


2. 分析识别系统的类及其关系，给出系统的面向对象设计，采用 UML 类结构图（ Class Diagram） 进行建模



3. 分析系统的行为，识别用例功能所对应的对象之间的交互场景，采用 UML 序列图（Sequence Diagram）进行建模

Host



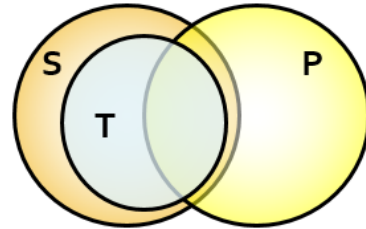
### 三、by 禹东（斜体表示注释）

#### 1、【黑白互补】

事实上，黑盒测试只能测到需求设计了的部分，白盒测试只能测到实际实现的部分。

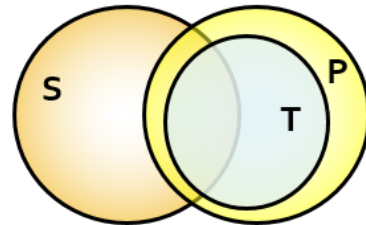
##### ■ Black-box testing

- Knowing the specified function that a product has been designed to perform
- Testing based on the interface
- Specification-based coverage



##### ■ White-box testing

- Knowing the internal workings of a product
- Testing based on the code
- Code-based coverage



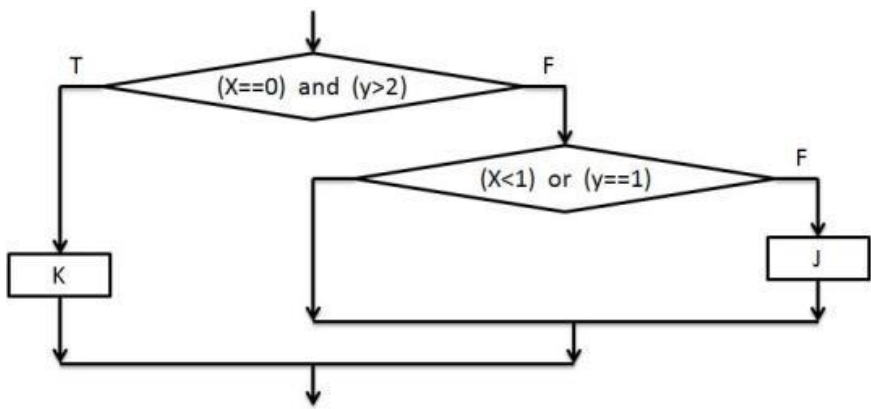
1) 如课件中计算 Next Date 的函数，假设函数鲁棒性不强，只对合法输入进行了处理。则在黑盒测试时可用 [2016-2-30]发现“函数对非法输入未做判断”的错误，但白盒测试只能覆盖所有正确输入的情况，无法发现这样的错误。

2) 仍以 NextDate 函数为例，假设黑盒测试用等价类测试法，分别测了几组非法输入（如[0-0-32]等）和一种正确输入（[2012-12-12]），未发现问题。但白盒测试中，为覆盖判断二月的分支，输入[2015-2-28]却得到[2015-2-29]，由此才发现代码的错误。

一个更极端的例子，假设在代码中有一段永远不会执行到、但会出错的代码：`while (false) {a = a / 0;}`，则黑盒测试永远无法检测到，但白盒测试在覆盖 `while` 语句的两个分支时则会检测到。

2、【白盒测试】

以题中的图为例，可解释几种覆盖如下：



语句覆盖：执行到J、K

判定（分支）覆盖：每个判断语句的两个分支都取到

条件覆盖：X==0、y>2、x<1、y==1 四个条件的真、假两种情况都取到

路径覆盖：覆盖每条路径至少一次

A：④路径覆盖 是最强的覆盖准则。

B：实现判定覆盖至少采用 ⑤I、II 和 III 组

C：实现条件覆盖至少采用 ④I 和 IV 组

3、【黑盒测试】

1) 覆盖五个输出

a	b	c	输出类型
1	1	1	正三角形
1	2	2	等腰三角形

3	4	5	不等边三角形
-1	1	1	a 的取值非法
1	2	3	非三角形

2)

等价类：对输入空间划分，c1~c3 各对应三个区间（ $x < 1$ 、 $1 \leq x \leq 200$ 、 $x > 200$ ），c4~c6 各对应两个区间（true、false），故可以把输入空间划分为 216 个等价类。下面仅取其中 5 个：

a	b	c
0	0	0
5	5	5
300	300	300
1	1	7
1	7	1

3) 边界值（就是对输入区间边界的情况单独测试）

a	b	c	边界情况
1	1	0	超出范围
200	200	200	合法边界
1	1	200	合法边界
200	201	200	超出范围
1	2	3	非三角形