OO博客

规格化设计是在结构化设计方法下应运而生的，他的目的是为了定义抽象对象的行为。用户会依赖这个行为，而实现着必须提供它。一个提供了所描述行为的实现称为可以满足规格。

规格的含义定义为满足其所有程序模块的集合。我们称之为规格的规格满足集。

为什么人们十分重视规格呢？规格对实现程序模块化是必不可少的。抽象用于将一个程序分解为多个模块。然而，抽象是无形的，没有描述，我们就无法知道他是什么或者如何将它与实现的抽象区分开来。而规格提供了这种描述。

规格对程序文件很明显是有用的。这个编写规格的行为也是很有意思的，因为它使正在规格的抽象清除显示出来。

规格化设计的核心要素主要有限制性，一般性以及清晰性。

* 如果一个规格排除了不能被抽象的用户接受的所有实现，则这个规格就是充分限制性的。
* 如果一个规格不排除可接受的实现，则该规格就是充分一般性的。
* 一个规格必须是清晰的，这样用户才容易理解。

对于过程规格来说，由于过程完成从输入到输出的转换计算所以可能会修改输入参数，会产生副作用以及可能会显式/隐式返回结果。并且过程的实现应该不受在何处使用的影响。过程规格抽象是描述过程执行后获得的效果（后置条件），而不是具体怎么做，不同的实现只是在细节方面有差异，并且一种实现可以被另一种实现替换。

过程规格的特性包括最少限度性，一般性，简单性。最小限度性即只强调使用者关心的要求，避免不确定性是指如果对于给定的输入，规格产生的结果依赖于具体实现，设计者应避免产生不确定性为。一般性是如果规格A比规格B能处理更多可能输入，则规格A更具有一般性。简单性是指规格应该保持简单，一个过程不应该做太多事情。

而如何设计类的规格呢？首先要定义类的规格，它包括类的目标和类的方法及其规格。Overview要告诉使用者,一个类管理/维护什么数据，提供哪些能力（不是简单列出有哪些方法）。接下来要

* 1. 定义抽象函数
  2. 定义表示不变式
  3. 实现类的属性

规格主要有两个用途，第一，通过将注意力集中于抽象和属性，编写规格的行为使被规格的抽象能清楚显示出来。我们应该自己关注那些可能被忽略的属性，包括requires格式中的应该声明的内容，确切的说，在应该用信号表明异常时，以及在处理限制问题时，关注这些可能被忽略的属性可以增强这个用途、这个用途有时候是规格的一个好处。

第二个用途是作为文件。规格在软件开发的每个阶段从设计到维护都很有价值。当人，它们不是所要求的唯一一个程序文件。规格描述了一个模块所做的内容，但任何一个实现很好的模块也都必须拥有解释它如何工作的文件。如果清楚的区别开来文件的这两种形式，则程序修改和维护会被简化。

对于我的作业来说，有一次因为没有清楚表明传入方法的变量不能为null，导致程序crash被算了4分，而我发现的别人的错误中，还没有与规格描述不清有关系的逻辑bug。

但是规格还是十分重要的。