软件构造 Blog-6 Lab3

写在最开始

在刚刚开学的时候，Mr.Wang说过这么一句话“我在五一假期给大家留了最难的那个实验，希望大家可以好好利用五一假期”，果然名不虚传，在发布了 这个实验之后，看到22页的实验手册就感觉这个实验不简单。今天实验刚刚验收完，虽然还有一部分附加内容没有写，仍然趁这个机会把过去两三周的 心路历程记录一下。

首先简要的说一下自己的看法吧，这个实验面向的是软件构造课程的第五章和第六章，面向复用和面向可维护性两个指标的实验，不可避免的需要对于 已有的问题进行分类和抽象，提取共性以实现面向复用性的指标；而对于面向可维护性，则需要在实验中更改已有的需求，然后来检验是否可以使用尽 量少的代码改动和遵守SOLID原则。这样在最初的时候如果对于所有的要求，（不包括3.12中对于变动的要求）没有一个全面的理解和抽象，那么在最 后就有可能能造成改动代码量非常大的情况。所以建议所有在开始写这个实验的时候，最好能够将实验手册上的所有要求都阅读一遍，以避免我刚刚提 的那种情况。

具体体会

简单的说一下我对于这个实验的实现思想和一些我在实验中遇到的问题。

- 首先是对于扩展 `Graph<L, E>`，这个接口不同于以往的，有两个泛型参数，分别是顶点和边的类型，这个跟以往的设计还是有差别的。感觉泛型虽 然不难，但是对于其中的一些细节还是要把握好，具体的后面再说。
- 其次就是两个抽象的顶点类和边类，其实说我们是可以针对所有的泛型参数的一个图。但是实际上由于我们Graph Interface中使用了一些特殊的性 质，只针对于基于那两个抽象的顶点类和边类所继承的子类才可能拥有其中的一些结构。
- 对于vertex包内对于顶点类的一些要求，我们可以在已有要求中抽象出来出于 `Vertex` 类和已有的类之间的一层中间的抽象类，比如 `Router, Server, Computer` 这三个类，它们的属性完全相同，那么其实我可以做一个进一步的抽象，这样可以规避在 `Network Topology` 里多模图处理的问， 使用这种抽象能够避免在其中直接使用 `Vertex` 的一些问题（例如增加顶点时不需要其余的附加条件，静态检查即可）。但是注意一个坑， `Person` 类 和 `Actor` 以及 `Director` 类属性完全相同，仅仅有性别和年龄的顺序不同，在抽象的时候却不能作为同一种类型去抽象。
- 对于Edge类及其子类，实验手册中已经做了官方的这种抽象，当然也可以继续抽象，但当时我没有想到，所以重构的时候可能会改一下。