

面向对象理解

一.为何要学面向对象?



对于一个需求的改变,为什么程序员的反响会这么大,究其原因改需求所要付出的时间,精力将是巨大的,所以我们现在的主流编程思想就成了面向对象.

第一,一个项目分成多个组去实现,分而治之,并行开发.很大程度上提升了开发的效率;

第二,是项目进行中,要是要修改或者增加某项功能,只要那一个模块变就可以.应变的能力强.

第三,是项目结束以后的维护,同样是因为模块化设计维护起来,会更方便,付出的成本更低.

可见,面向对象对于程序员来说是非常有利的,也是所憧憬的,难怪屹立这么多年都没倒下.

二.对象和类

2.1 对象

万物皆对象,所以组成世界的所有物质都是对象,而什么又是物质,哲学上说,物质指不依赖于人们的意识而存在,又能为人们的意识所反映的客观实在.说白了对象就是我们见到的一切东西.

2.2 类

对象的抽象,为什么抽象一个类出来呢,其实是为了让我们更好的认知.所以说面向对象是符合人类的思维方式的.而在编程中,也是同样的道理,三大特征,六大原则已经很好的展现了其优势.

那么,接下来让我们来用面向对象的角度看一看这次答辩:

谁?干嘛?

我们,答辩...

三.特征:

3.1 封装:

3.1.1 数据角度

我们每个人都把自己千丝万缕的思想用文字这种基本类型合成了了一个自定义的类型, 名为文档. 不得不说非常符合人类的思考方式.

3.1.2 功能角度

我们每个人, 上台只对外提供我们大家对面向对象的理解这一必要的功能, 而隐藏了我们怎么千辛万苦准备这一切所实现的细节..

3.1.3 设计角度

a. 分而治之: 我们马老师负责监督我们答辩, 我们负责准备答辩. 需求不同, 协同完成答辩这一大事.

b. 封装变化: 因为师生职责不同这个变化点, 所以在此次答辩中, 我们跟马老师其实就是两个类. 要单独封装. 并且在答辩的时候进行了交互.

c. 高内聚: 我们真的都很用心的在准备! 就是为了能够通过最好的状态上台讲述自己的理解.

d. 低耦合: 我们跟马老师的关系, 不会出现任何逾距的情况, 互不影响, 只有在必要的时候进行交互.

3.2 继承

以我们所有学生对象来说抽象出一个学生类, 那么, 新来的同学, 就相当于是一种扩展, 并且重用了学生类的概念. 所以讲到这里, 还是用泛化更利于个人的理解, 因为我们会发现, 这其实是首先从具体到抽象, 而在有了抽象的概念之后才有了扩展这一说.

3.3 多态

简单的说就是调父执行子, 我们每一位都可以看成父类(学生类)中的一个子类. 每一次的被马老师调用上台述说, 我们的实现方式都不一样, 这就是多态. 马老师关心的是有学生上台去讲, 最后讲完了, 大家对面向对象都有了更深的理解, 那么这次活动的目的就达到了, 至于说, 指定谁先谁后并不重要.

四.六大原则:

4.1 开闭原则

这次答辩,完全可以增加一个新的内容,比如聊聊人生,这个时候,马老师依旧调用的方法是让学生上台述说,而我们也不需要把有关的面向对象的内容删掉.,而只是增加一节内容,聊聊人生而已.

4.2 类单一职责

我们学生类只负责怎么好好想想如何答辩,而改变我们是否要的答辩原因只有一个,我们到底想不想学好面向对象.

4.3 依赖倒置

马老师找学生上台讲述面向对象,调用的是抽象的我们,而不是具体的某个谁,因为马老师对我们每个人都一视同仁.

4.4 组合复用

前面那么多同学都已经讲述过了,那么我们后面的同学再讲的时候,一定要注意,一定要优先选择组合复用,而不是继承.要尽可能耦合性要低些,避免出现简单的重复.

4.5 里式替换

我们同学类无论谁都有可以代替我们此次上台的答辩学生类,答辩本质没有变.

4.6 迪米特法则

体现的就是低耦合,这场答辩,马老师跟我们的交流很少,也应该是少的,这样,我们才能发挥每个人的主观能动性去自己总结,这样,我们每个人的表达方式不一样,但是并不会影响马老师对我们答辩 的关注.

五.三种关系:

5.1 泛化关系

从我们每个学生到学生类,就是一种泛化.

5.2 关联关系

我们本身与我们的手稿体现的就是一种关联的关系,因为手稿出自我们自己的手,我们每个人可以说成,我有一份手稿.

5.3 依赖关系

我们依赖投影机,幕布这些设备,来协助完成我们的答辩.

以上,即我对面向对象的理解,如有雷同,纯属巧合.

Aid1903

徐樟兵