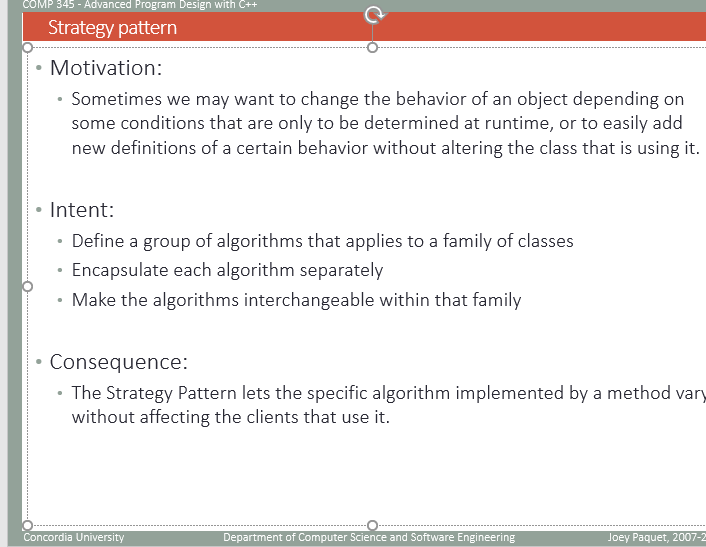
strategy



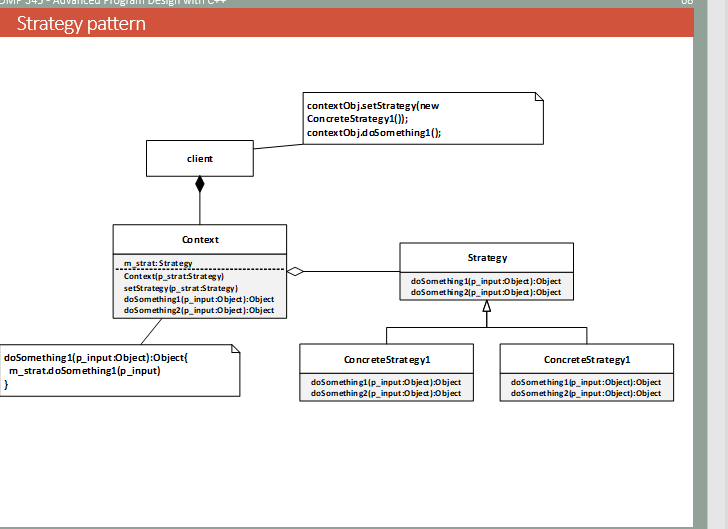
动机就是改变一个object的function的具体行动，取决于在runtime时决定的condition，或者说能够在不改变class的情况下添加新behavior

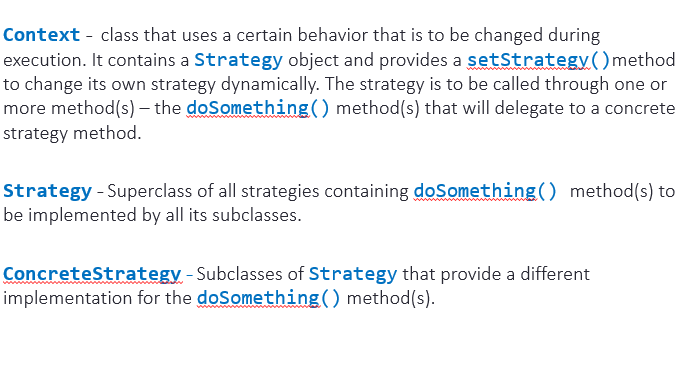
怎么做：

描述一组算法

将这些算法分别封装

让这些算法



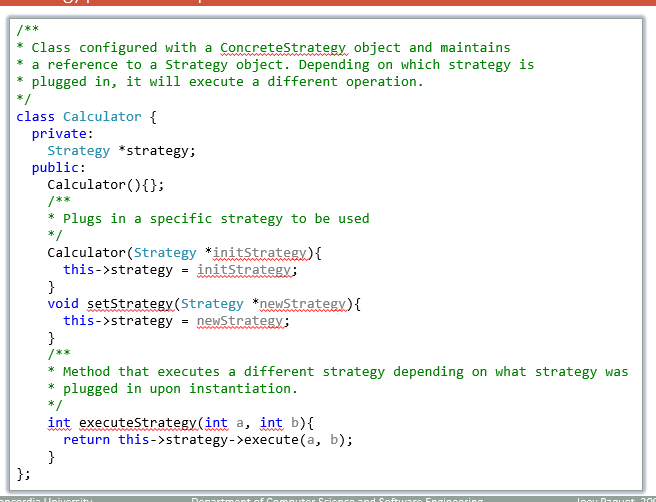
Context:主体：拥有一个Strategy object与SetsTRATEGY

然后他的constructor必须有strategy参数

doSomething就是让我们的strategy进行它内部的execute

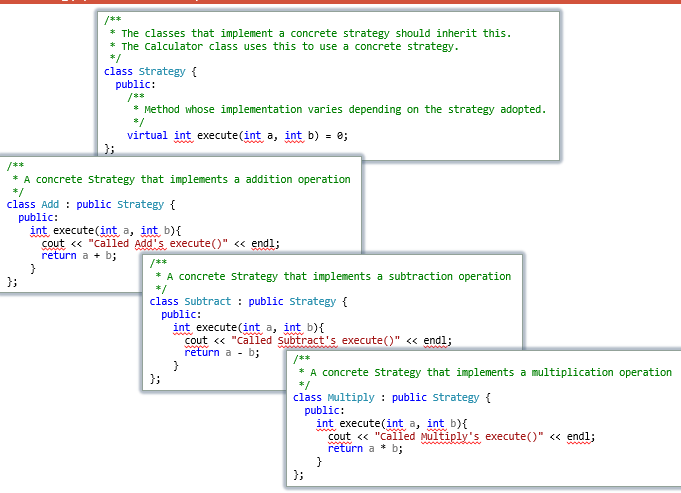
下图的executeStrategy,会

return strategy抽象类的execute



executeStrategy的具体效果取决于是啥strategy

strategy，所有具体strategy的父类，拥有doSomething method



Strategy是父类

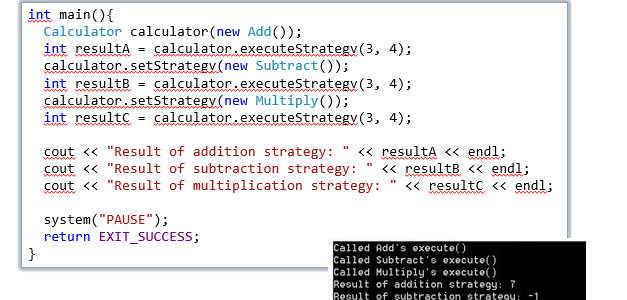
有一个等待被描述的virtual接口

add

subtract

multiply

这三个是具体行为



你可以看见虽然参数都是一样的，但因为call了不同的strategy结果不一样