**类爆炸**

“类爆炸”是面向对象设计中的一个问题，指的是在设计和实现系统时，由于引入了大量的类来处理不同的需求和变化，导致类的数量迅速增长。这种情况会带来复杂性和维护困难。以下是详细解释：

**类爆炸的成因**

需求变化频繁：每次需求变化时，开发者可能会创建新的类来满足新的需求，而不是对现有类进行重构。

过度继承：滥用继承关系，不合理地将多个子类派生自一个父类，导致子类数量急剧增加。

功能单一：为了遵守单一职责原则，每个类只负责一个功能，但没有合理设计抽象和组合，导致类的数量和层次过多。

低内聚高耦合：类的设计缺乏良好的封装和模块化，使得类之间的依赖关系过于复杂。

**类爆炸的后果**

维护困难：大量的类会增加系统的复杂性，使得维护变得更加困难。开发者需要花费更多的时间来理解和修改代码。

性能问题：过多的类可能会导致内存和性能问题，尤其是在高频调用和大规模并发的场景中。

代码冗余：类似功能的类可能会出现代码冗余，增加了代码库的体积和维护成本。

**解决类爆炸的方法**

合理设计抽象：通过抽象类和接口，设计合理的层次结构，减少不必要的类。

使用设计模式：采用设计模式，如策略模式、装饰者模式等，来减少类的数量并提高代码的复用性和灵活性。

重构现有代码：定期进行代码审查和重构，合并重复的类和功能，简化系统结构。

模块化设计：将系统划分为多个独立的模块，每个模块内部使用合适的设计模式，降低类的数量和复杂度。

《大话设计模式》

1. 计算器控制台程序，要求输入两个数和运算符号，得到结果【简单工厂，工厂方法】
2. 商场收银，商场促销【策略】
3. 简历复印【原型】
4. 购买基金【外观】
5. 画人【建造者】
6. 工作状态【状态模式】
7. 翻译者【适配器】
8. 分公司【组合】
9. 上车买票【迭代器】（不管什么都得遍历一遍）
10. 不同的手机品牌都有通讯录、游戏【桥接】
11. 加薪审批【职责链】