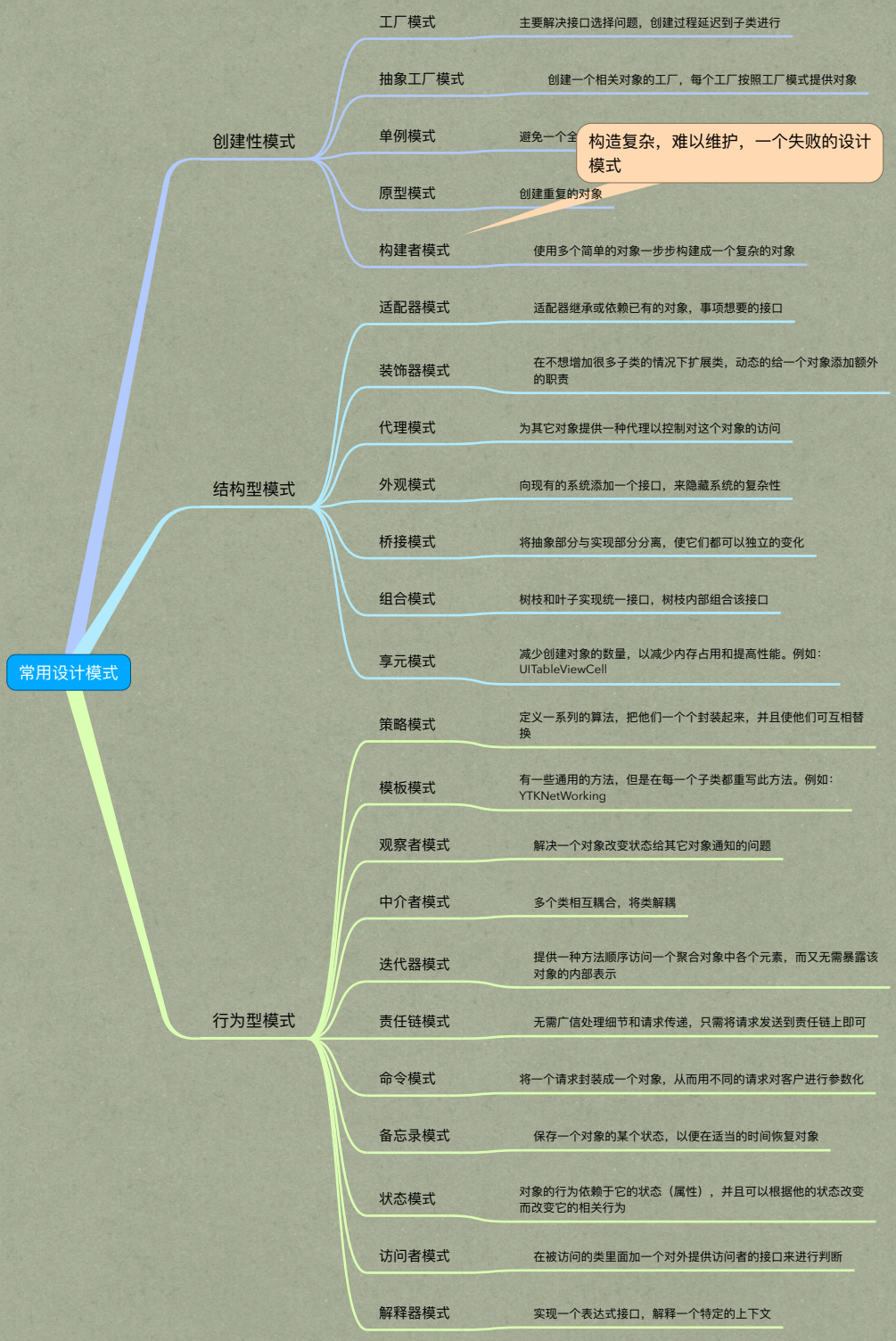
常用设计模式
创建性模式
工厂模式
主要解决接口选择问题，创建过程延迟到子类进行
抽象工厂模式
创建一个相关对象的工厂，每个工厂按照工厂模式提供对象
单例模式
避免一个全
构造复杂，难以维护，一个失败的设计模式
原型模式
创建重复的对象
构建者模式
使用多个简单的对象一步步构建成一个复杂的对象
结构型模式
适配器模式
适配器继承或依赖已有的对象，事项想要的接口
装饰器模式
在不想增加很多子类的情况下扩展类，动态的给一个对象添加额外的职责
代理模式
为其它对象提供一种代理以控制对这个对象的访问
外观模式
向现有的系统添加一个接口，来隐藏系统的复杂性
桥接模式
将抽象部分与实现部分分离，使它们都可以独立的变化
组合模式
树枝和叶子实现统一接口，树枝内部组合该接口
享元模式
减少创建对象的数量，以减少内存占用和提高性能。例如：UITableViewCell
行为型模式
策略模式
定义一系列的算法，把他们一个个封装起来，并且使他们可互相替换
模板模式
有一些通用的方法，但是在每一个子类都重写此方法。例如：YTKNetWorking
观察者模式
解决一个对象改变状态给其它对象通知的问题
中介者模式
多个类相互耦合，将类解耦
迭代器模式
提供一种方法顺序访问一个聚合对象中各个元素，而又无需暴露该对象的内部表示
责任链模式
无需广信处理细节和请求传递，只需将请求发送到责任链上即可
命令模式
将一个请求封装成一个对象，从而用不同的请求对客户进行参数化
备忘录模式
保存一个对象的某个状态，以便在适当的时间恢复对象
状态模式
对象的行为依赖于它的状态（属性），并且可以根据他的状态改变而改变它的相关行为
访问者模式
在被访问的类里面加一个对外提供访问者的接口来进行判断
解释器模式
实现一个表达式接口，解释一个特定的上下文