

基本模型：（Java，C#，Scala）

1. 接口，类
2. 接口可以继承多个接口
3. 类可以实现多个接口
4. 类只能继承一个父类

扩展模型一：静态扩展（C#，Swift）

1. 静态扩展通常属于项目后期的一种行为。
2. 静态扩展针对特定的类型，可重用性不佳。
3. Extension可以扩展一个类，为类添加新的方法。

扩展模型二：动态组合（Scala）

1. Trait类似于一个带全部或部分实现的接口。
2. Trait的设计是属于项目前期明确的设计行为。
3. Trait是一个独立的结构，被设计用来支持重用。
4. 在声明一个类时，Trait可以静态混入一个类的定义中。
5. 在实例化一个类时，Trait可以动态的混入到一个类的实例中。
6. 在Trait之前由于不支持实现的多继承，因此如果一个类要使用其它类的功能，就需要实例化一个其它类的实例，这相当于在运行过程中动态的把两个类的功能组合起来了。使用Trait之后则可以以声明的形式来组合代码，这样写出的代码更具可读性，而且如果有问题在编译阶段就可以发现。