

类簇

[编辑](#)

 本词条缺少概述、信息栏、名片图，补充相关内容使词条更完整，还能快速升级，赶紧来[编辑](#)吧！



收藏

5

1

词条统计

浏览次数：1176次

编辑次数：4次 [历史版本](#)

最近更新：2013-10-07

创建者：[inroading](#)



百科消息：

- 【数字博物馆】观复博物馆入驻百科
- 中国少年男子组合TFBoys成军一周年
- 【詹天佑故居上线】还原真实的詹天佑
- 赤火虹口革命地，中共四大纪念馆上线
- 【百科新玩法】我来建设琼中城市百科

1 概述

 [编辑](#)

类簇是Foundation框架中广泛使用的设计模式。类簇将一些私有的、具体的子类组合在一个公共的、抽象的超类下面，以这种方法来组织类可以简化一个面向对象框架的公开架构，而又不减少功能的丰富性。

2 在类簇中创建子类

 [编辑](#)

类簇架构是简洁性和可扩展性之间的一个折衷。用几个公共类代表多个私有类可以使框架中的类易学易用，但是在某种程度上增加创建类簇子类的难度。然而如果很少需要创建子类，则类簇架构则有明确的好处。Foundation框架只在这种情况下使用类簇。

如果您发现某个类簇没有提供您的程序需要的功能，则可能适合引入一个子类。举例来说，假定您希望创建一个数组对象，其存储是基于文件的，而不是如NSArray类簇那样是基于内存的。由于您需要改变类的存储机制，所以需要创建子类。

另一方面，在某些情况下可能定义一个拥有类簇对象的类就够了（而且更加容易）。假定您的程序需要在某些数据被修改的时候得到通知，则为Foundation框架中定义的数据对象创建一个简单的包装类可能是最好的方法。该类的对象可以干预修改数据的消息，将它截获并进行必要的动作，然后再转发给嵌入的数据对象。

总的来说，如果您需要管理对象的存储，就创建一个真的子类。否则，就创建一个合成对象，即将标准的Foundation框架对象嵌入到您自己设计的对象中。

3 类簇的概念

 [编辑](#)

类簇 是一寸隐藏在通用接口下的与实现相关的类，使得我们编写的代码可以独立于底层实现（因为接口是稳定的）。

如创建NSString对象时，你得到的可能是NSLiteralString,NSCFSString,NSSimpleCString等。即不同的NSString对象调用同一个接口A，接口A的实现可能是不同的。

在Foundation框架中，常见的类簇有NSString,NSArray,NSDictionary等。 想要在类簇中创建子类会困难一些，必须是抽象超类的子类，必须重载超类的原始方法，必须声明自己的数据存储。最方便的是使用组合或者类别来代替子类化。

4 类簇架构代表一个反映简单概念的设计。

 [编辑](#)

类簇中的抽象超类必须声明创建其私有子类的方法。根据调用的创建方法分配正确类型的对象是超类的责任。在Foundation框架中，通常通过调用+ className...或alloc...和init...方法来创建对象。以Foundation框架的NSNumber类为例，可以发送如下的消息来创建数字对象：

NSNumber *aChar = [NSNumber numberWithIntChar:'a'];
NSNumber *anInt = [NSNumber numberWithInt:1];
NSNumber *aFloat = [NSNumber numberWithFloat:1.0];
NSNumber *aDouble = [NSNumber numberWithDouble:1.0];



新手上路

成长任务

编辑入门

编辑规则

百科术语



我有疑问

常见问题

我要提问

参加讨论

意见反馈



投诉建议

举报不良信息

未通过词条申诉

投诉侵权信息

封禁查询与解封