1. 分类和扩展有什么区别？可以分别用来做什么？分类有哪些局限性？分类的结构体里面有哪些成员？
2. ①类别中原则上只能增加方法（能添加属性的的原因只是通过runtime解决无setter/getter的问题而已）；  
   ②类扩展不仅可以增加方法，还可以增加实例变量（或者属性），只是该实例变量默认是@private类型的（  
   用范围只能在自身类，而不是子类或其他地方）；  
   ③类扩展中声明的方法没被实现，编译器会报警，但是类别中的方法没被实现编译器是不会有任何警告的。这是因为类扩展是在编译阶段被添加到类中，而类别是在运行时添加到类中。  
   ④类扩展不能像类别那样拥有独立的实现部分（@implementation部分），也就是说，类扩展所声明的方法必须依托对应类的实现部分来实现。  
   ⑤定义在 .m 文件中的类扩展方法为私有的，定义在 .h 文件（头文件）中的类扩展方法为公有的。类扩展是在 .m 文件中声明私有方法的非常好的方式。
3. 最重要的还是类扩展是在编译阶段被添加到类中，而类别是在运行时添加到类中。  
   分类方法未实现，编译器也不会报警告。  
   分类方法与原类中相同会优先调用分类。
4. 分类的结构体
5. typedef struct objc\_category \*Category;
6. struct objc\_category {
7. char \*category\_name OBJC2\_UNAVAILABLE; // 分类名
8. char \*class\_name OBJC2\_UNAVAILABLE; // 分类所属的类名
9. struct objc\_method\_list \*instance\_methods OBJC2\_UNAVAILABLE; // 实例方法列表
10. struct objc\_method\_list \*class\_methods OBJC2\_UNAVAILABLE; // 类方法列表
11. struct objc\_protocol\_list \*protocols OBJC2\_UNAVAILABLE; // 分类所实现的协议列表
12. }