[**objective-c基础教程——学习小结**](http://www.cnblogs.com/lm3515/archive/2010/12/20/1911135.html)

**提纲：**

简介

与C语言相比要注意的地方

objective-c高级特性

开发工具介绍(cocoa 工具包的功能，框架，源文件组织；XCode使用介绍)

**简介：**

1.       objective-c是C语言的一个扩展集，主要由APPLE公司维护，是MAC系统下的主要开发语言。

个人认为，对于用惯了常用的C,JAVA等语言的人来说，objective-c是一中很另类，非主流的语言。

2.    开发 Mac的 UI 程序来说，使用的是 Cocoa 这个框架，cocoa的组成部分有：foundation和application kit框架。

【foundation框架处理用户界面之下的特性，如数据结构和通信机制；application kit框架包含cocoa的高级特性：用户界面元素，打印，颜色，声音管理，applescript等】

3.       我通过这本书的学习：

基本掌握了Objective C的语法，基本能看懂别人写的代码，自己也能编写代码；

熟悉了开发环境XCode的使用；（包括建立项目，调试，运行，代码管理等）

**与C语言相比要注意的地方：**

1.       文件介绍：Objective-C 也使用头文件(header files)，后缀为 .h, 但使用 .m（即 message, 其他面向对象编程语言也叫 method），作为源文件的后缀。

在objective-c中使用＃import<>,而不使用＃include<>，＃import可以保证头文件只被包含一次。

2.       与C一致的地方：

数据类型，表达式，各种运算符

循环：for, while, do while, break, continue

分支：if, else, switch

3.    NSlog()函数：与printf()类似，向控制台输出信息。但它增加了一些特性，如时间戳等。

【cocoa对起所有的函数，常量和类型名称都添加了NS前缀。】

4.       双引号的前面的@表示这双引号中的字符串应该作为cocoa的NSString元素来处理。

5.       布尔类型BOOL:值为YES,NO;

6.    %d表示输出整数

  ％@表示输出NSString类型

%s表示输出字符串数组（char\*）;

7.    objective-c中的方括号：

1.       用于通知某个对象该做什么。

2.       方括号内第一项是对象，其余部分是你需要对象执行的操作。

3.       在objective-c中通知对象执行某种操作，称为**发送消息**。（也叫调用方法）

8. 标识符id:是一种泛型，用于表示任何种类的对象。

9. 类声明@interface：

|  |
| --- |
|  |
| @interface Circle:NSObject   //为Circle类定义接口；NSObject表示是父类。  {//以下是数据成员     ShapeColor fillColor;     ShapeRect bounds;  }  //以下是方法声明  -(void) setFillColor: (ShapeColor) fillColor;  //前面的短线表明这是方法声明                                     //短线后面是方法的返回类型                                     //接着是方法名  //冒号后面是参数，其中(ShapeColor)是参数类型，  fillColor是参数名  -(void) setBounds:(ShapeRect) bounds;  -(void) draw;   //减号就是普通函数  加号就是静态函数  @end     //结束声明 |
| objective-c看到@符号，就把它看成是C语言的扩展。 |
| **一个类的完整声明如下：**      @interface CClassA(Category): CBaseClass<IA,IB...>  //类名（类别名）：父类<协议> |

10. 类实现@implementation

|  |
| --- |
| @implementation |
| @implementation Circle  -(void) setFillColor: (ShapeColor) c  {     fillColor=c;  }  -(void) draw  {      NSLog(@”drawing a circle at(%d %d %d %d) in %@”, bounds.x,bounds.y,bounds.width,bounds.height,colorName(fillName));  }  @end |
| 隐藏对象self对应于C中的this。  Self->fillcolor来访问成员变量。 |

11. 调用写好的类和类函数：

|  |
| --- |
| //创建新对象，使用缺省初始化函数      Bank \*bankDefault = [[Bank alloc] init];  //调用方法：      [bank addAmount: 1];  [bank print];  // 释放对象：      [bankDefault free]; |
|  |

12.

|  |
| --- |
| 两个参数的方法： |
| -(void) setTire: (Tire \*) tire  //声明        atIndex: (int) index;  //使用  [car setTire:tire atIndex:2]; |

**objective-c高级特性：**

1.       继承

|  |
| --- |
| Objective-c不支持多继承。 |
| Super 关键字：调用该类的父类；  超类：父类的另一种说法。 |
|  |

2.

|  |
| --- |
| 自定义NSLog()输出: |
| 在类中添加description方法就可以自定义NSLog()如何输出对象。 |
| @implementation Tire  -(NSString \*) description  {     Return (@”I am a tire.”);  }  Main()  {      NSLog(@”%@”,tire[0]);  } |

3.

|  |
| --- |
| **Foundation kit:** |
| Cocoa是由两部分框架组成的：foundation kit [包括一些基础类]和 application kit.【包括用户接口对象和高级类】 |
| 创建字符串：  NSString \*test;  test=[NSString stringWithFormat:@"i'm %d years old!",23]; |
| 若在声明方法时在方法前面添加了加号，那就表示把这个方法定义为类方法【这个方法属于类对象，而不是类的实例对象。】 |
| NSArray类：可以存放任意类型的对象.  它有两个限制：  1. 它只能存储objective-c的对象，但不能存储C中的基本数据类型，如int , float, enum, struct等。  2.不能存储nil(对象的零值或NULL值)；【因为在创建NSArray时，要在列表结尾添加nil代表列表结束。】  创建NSArray：  NSArray \*array;  array=[NSArray arrayWithObjects:@"one",@"two",nil]; |
| NSString, NSMutableString类；  【NSString是不可变的，即一旦创建完成，就不能通过删除字符或添加字符的方式来改变它；  而NSMutableString是可变的。  这两个类就像JAVA中的string与stringBuffer类的区别。】  NSArray, NSMutableArray类；  NSEnumerator枚举；  【  NSEnumerator \*emun;  Emun=[array objectEnumerator];  Id thingie;  While(thingie=[enumerator nextObject]){}  】  NSDictionary:字典（关键字及其定义的集合。）【也成为散列表，关联数组】，NSMutableDictionary类；  NSNumber:用来包装基本数据类型，如int ,char, float, bool;【将一个基本类型的数据包装成对象叫做装箱。】  NSValue:它可以包装任何类，NSNumber是它的子类。  NSNull: |
| 在cocoa中看到“CF”字样时，就表示它是苹果公司的Core Foundation 框架相关的内容。 |
| NSAutoreleasePool:自动释放内存池。 |
|  |

4.

|  |
| --- |
| 内存管理 |
| 每个对象都有一个与之关联的引用计数（也叫保留计数） |
| 当使用alloc, new 方法或通过 copy消息（生成接收对象的一个副本）创建一个对象时，对象的引用计数值被设为1；   给对象发retain消息时，增加该值；   发送release消息时，减少该值；  当一个对象的引用计数值变为0时，objective-c会自动向对象发送一条dealloc消息。销毁该对象。  你可以在自己的对象中重写该方法，  使用retainCount消息，可以获取引用计数器的值。  -(id) retain;  -(void) release;  -(unsigned) retainCount; |
| 自动释放池：autorelease pool;  创建：  NSAutoreleasePool \*pool;  pool=[[NSAutoreleasePool alloc] init];  销毁：  [pool release];  注意：xcode自动生成的代码，销毁pool池时，使用的是[pool drain],drain方法只是清空释放池，但不销毁pool.所以在自己编写代码时还是使用release.  而且，drain只适用于MAC OS 10.4以上的版本，而release适用于所有版本。  只有在向某个对象发送autorelease消息时，该对象才会添加到NSAutoreleasepool中，才会被自动释放。  如：[car autorelease]; |
| **内存管理黄金准则：**  只有通过alloc, new和 copy方法创建的对象，才需要程序员负责向该对象发送release或autorelease消息。  而通过其他方法获得的对象，则默认为已经被设置为自动释放，所以不需要程序员做任何操作了。 |
| 在objective-c 2.0中有垃圾回收机制，  如果要对某个项目使用垃圾回收：  项目信息--build选项卡--查询"garb",出现“objective-c Garbage Collection”,将其值设置为“required[-fobjc-gc-only]”  启用垃圾回收后，通常的内存管理命令全都变成了空操作指令，不执行任何操作。  开发iphone软件，不能使用垃圾回收。 |
|  |

5.

|  |
| --- |
| 对象初始化 |
| 两种创建新对象的方法：  [类名 new]  [[类名 alloc] init]  这两种方法是等价的，但cocoa的惯例是使用后者。 |
| alloc在为对象分配空间的同时，将这块内存初始化为0； |
| Init方法：初始化实例变量，使对象处于可用状态。[返回类型为id,  返回的值描述了被初始化的对象]  使用new创建新对象时，系统要完成两个步骤：  1.       为对象分配内存，即对象获得一个用来存放其实例变量的内存块；  2.       自动调用init方法，让该对象处于可用状态。 |
|  |
|  |

6.

|  |
| --- |
| objective-c 2.0的新特性【只适用于mac os x10.5及以上】 |
| @property :表示声明了对象的属性。【这样就不用再写属性的访问器了。】  （他有copy, retain, readwrite, readonly等属性） |
| @synthesize:表示“创建该属性的访问器” |
| 点表达式 |
|  |
|  |

7.

|  |
| --- |
| 类别 |
| 类别（category）是一种为现有的类添加新方法的方式。 |
| 类别的声明：  @interface NSString (NumberConvenience) //类名 （类别名）  -(NSNumber) lengthAsNumber;            //扩充方法声明  @end  使用时使用原来的类名，就可以调用他的所有类别中的方法。 |
| 类别的局限性：  1.    不能向类中添加新的实例变量；  2.    在类别中的方法若与类中现有的方法重名，则类中的方法不可用，被类别中的新方法取代。 |
| 类别的作用：  1.    将类的实现分散到多个文件或框架中；  2.    创建对私有方法的前向引用；  【Cocoa中没有真正的私有方法，则实现私有方法类似功能的方法为：  先在类别中声明方法；然后到现有类的实现中实现该方法。  这样这个类中的其他方法可以使用该方法，而其他外部的类就不会知道该方法的存在了。】  3.    向对象添加非正式协议。  【创建一个NSObject的类别称为创建一个非正式协议。】 |
| 委托delegate是一种对象，另一个类的对象会要求委托对象执行它的某些操作。  **受委托对象在某个时间（某个事件触发）时，会自动通知委托对象执行委托方法。** |
| **选择器：@selector（）**:选择器只是一个方法名称，但它以objective-c运行时使用的特殊方式编码，以快速执行查询。圆括号中的内容是方法名。  所以Car类的setEngine:方法的选择器是：@selector（setEngine:  受委托对象如何知道其委托对象是否能处理它（受委托对象）发送给它（委托对象）的消息？  通过选择器，受委托对象先检查委托对象，询问其是否能响应该选择器。如果能，则向它发送消息。 |
|  |

8.

|  |
| --- |
| 协议： |
| 正式协议是一个命名的方法列表。  采用协议意味着必须实现该协议的所有方法。否则，编译器会发出警告。  **正式协议就像JAVA中的接口一样。** |
| 声明协议：  @protocal NSCopying  -(id) copywithzone:(NSZone \*) zone; //方法列表  @end |
| 采用协议：  @interface Car:NSObject <NSCopying,NSCoding> //中括号中是要实现的协议列表  {//实例变量列表}  //方法列表  @end |
| 在objective-c 2.0中，有新特性：@optional, @required |

9.

|  |
| --- |
| AppKit: |
| 创建APPKIT项目：  File—new project—mac os—application—Cocoa application; |
| IBOutlet与IBAction  这两个都是APPKIT提供的#defines.  IBOutlet没有任何作用，不会对它进行编译。  IBAction定义为void。  这两个是为Interface Builder以及阅读代码的人提供的标记。 |
| .xib文件一般称为.nib文件。  .nib文件是包含被冻结对象的二进制文件。  而.xib文件是XML格式的nib文件。 |