

GeekBand 极客班

互联网人才 + 加油站!



C++系统工程师



iOS开发工程师



Android开发工程师



PM产品经理

设计模式二

孔祥波

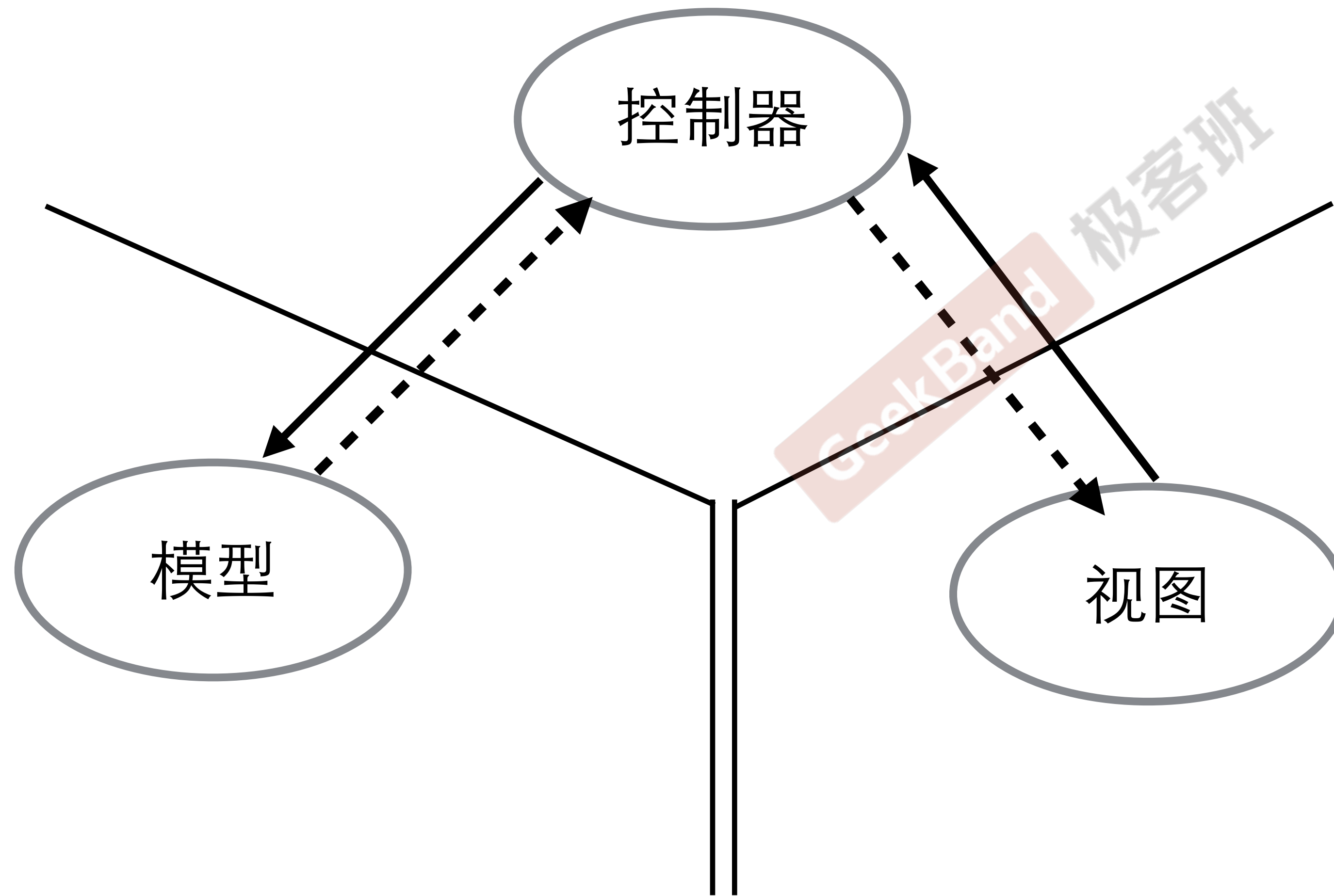
GeekPanda

“The best way to predict the future is to invent it.”

–Alan Kay

两步创建

Model(模型), View(视图), Controller(控制器)



两步创建

- alloc 和init (对象分配内存和初始化)
- 内存分配:Cocoa从应用程序的虚拟内存中为对象分配一块内存(Class/实例变量计算, 确定大小)
- alloc or allocWithZone: 发送消息, 获得一个实例
 - 对象的保持(retain)数设置为1.
 - 分配的对象的isa指针指向类对象。
 - 把对象所有的实例变量初始化为0.也可以理解成0的等价类型: nil

初始化对象

- 固定格式，一般initWithArguments:

```
@interface UIView : UIResponder
- (instancetype)initWithFrame:(CGRect)frame ;
- (instancetype)initWithCoder:(NSCoder *)aDecoder;
```

- 初始化可能失败
- 对象不能重复初始化，失败后要销毁(NSInvalidArgumentException)

初始化方法需要遵循以下步骤

- 1 先要调用父类的初始化方法
- 2 检查父类初始化返回的对象，如果是nil则初始化失败，也返回nil
- 3 在初始化实例变量时，如果它们与其他对象的引用，必要时要进行retain和copy
- 4 如果返回一个已存在的对象，那首先释放新分配的对象（刚才提到的账号的类）
- 5 遇到问题初始化不成功（比如初始化文件失败），返回nil
- 6 如果没有问题，返回self。初始化完成

注意：子类初始化时，必须先调用父类的初始化方法，以保证继承链中父类的实例变量得到正确的赋值。

其他

- dealloc , dealloc和init (malloc,free),分配内存/释放内存
- 工厂方法:
+ (id)dateWithTimeIntervalSinceNow:(NSTimeInterval)secs; (2部合为一个方法)
- 内省(Introspection),面向对象语言和环境的重要特性(runtime)
class,superclass,isKindOfClass,respondsToSelector

模板方法

GeekBand 极客班

模版方法

- 用一个可以重用的类实现通用的步骤
- 需要在子类中重写
- -dealloc 每个NSObject在没有使用ARC/垃圾自动回收，需要实现,释放资源
- UIView -drawRect

example

```
- (void)drawRect:(CGRect)rect {  
    CGRect bounds = [self bounds];  
    [[UIColor grayColor] set];  
    UIRectFill (bounds);  
    CGRect square = CGRectMake (10, 10, 50, 100);  
    [[UIColor redColor] set];  
    UIRectFill (square);  
    [[UIColor blackColor] set];  
    UIRectFrame (square);  
}
```

单例模式 singleton

对象的引用



单例模式

- 声明一个单例对象的静态实例，并初始化为nil。
- 在该类的类工厂方法（名称类似于“sharedInstance”或“sharedManager”）中生成该类的一个实例，但仅当静态实例为nil的时候。
- 重载allocWithZone:方法，确保当用户试图直接（而不是通过类工厂方法）分配或初始化类的实例时，不会分配出另一个对象。
- 实现基本协议方法：copyWithZone:、release、retain、retainCount、和autorelease，以保证单例的状态。


```
static MyGizmoClass *sharedGizmoManager = nil;

+ (MyGizmoClass*)sharedManager
{
    @synchronized(self) {
        if (sharedGizmoManager == nil) {
            [[self alloc] init]; // assignment not done here
        }
    }
    return sharedGizmoManager;
}

+ (id)allocWithZone:(NSZone *)zone
{
    @synchronized(self) {
        if (sharedGizmoManager == nil) {
            sharedGizmoManager = [super allocWithZone:zone];
            return sharedGizmoManager; // assignment and return on first allocation
        }
    }
    return nil; //on subsequent allocation attempts return nil
}

- (id)copyWithZone:(NSZone *)zone
{
    return self;
}
```

案例

- [UIApplication sharedApplication]
- [NSNotificationCenter defaultCenter]
- [NSUserDefaults standardUserDefaults]
- [NSFileManager defaultManager]
- [UIAccelerometer sharedAccelerometer]
- [NSURLSession sharedSession]

design

- 全局变量
- 公用组件http api



小结

- 两步创建
- 模版方法
- 单例模式

