
1. #import 跟#include、@class 有什么区别? #import<>跟 #import""又有什么区别?

- 1> #import 和#include 都能完整地包含某个文件的内容, #import 能防止同一个文件被包含多次
- 2> @class 仅仅是声明一个类名, 并不会包含类的完整声明;@class 还能解决循环包含的问题
- 3> #import <>用来包含系统自带的文件, #import ""用来包含自定义的文件

2. 属性 readwrite, readonly, assign, retain, copy, nonatomic 各是什么作用, 在那种情况下用?

- 1> readwrite: 同时生成 get 方法和 set 方法的声明和实现
- 2> readonly: 只生成 get 方法的声明和实现
- 3> assign: set 方法的实现是直接赋值, 用于基本数据类型
- 4> retain: set 方法的实现是 release 旧值, retain 新值, 用于 OC 对象类型
- 5> copy: set 方法的实现是 release 旧值, copy 新值, 用于 NSString、block 等类型
- 6> nonatomic: 非原子性, set 方法的实现不加锁 (比 atomic 性能高)

3. 写一个 setter 方法用于完成@property (nonatomic,retain) NSString *name, 写一个 setter 方法用于完成@property (nonatomic, copy) NSString *name.

```
1> @property (nonatomic, retain) NSString *name;
- (void)setName:(NSString *)name
{
    if (_name != name) {
        [_name release];
        _name = [name retain];
    }
}
2> @property (nonatomic, copy) NSString *name;
- (void)setName:(NSString *)name
{
    if (_name != name) {
        [_name release];
        _name = [name copy];
    }
}
```

4. 对于语句 `NSString*obj = [[NSData alloc] init];` , 编译时和运行时 `obj` 分别是什么类型?

- 1> 编译时是 `NSString` 类型
- 2> 运行时是 `NSData` 类型

5. 常见的 `object-c` 的数据类型有那些, 和 `C` 的基本数据类型有什么区别?

- 1> 常用 OC 类型: `NSString`、`NSArray`、`NSDictionary`、`NSData`、`NSNumber` 等
- 2> OC 对象需要手动管理内存, `C` 的基本数据类型不需要管理内存

6. `id` 声明的变量有什么特性?

`id` 声明的变量能指向任何 OC 对象

7. `Objective-C` 如何对内存管理的,说说你的看法和解决方法?

- 1> 每个对象都有一个引用计数器, 每个新对象的计数器是 1, 当对象的计数器减为 0 时, 就会被销毁
- 2> 通过 `retain` 可以让对象的计数器+1、`release` 可以让对象的计数器-1
- 3> 还可以通过 `autorelease pool` 管理内存
- 4> 如果用 `ARC`, 编译器会自动生成管理内存的代码

8. 内存管理的几条原则是什么? 按照默认法则.哪些方法生成的对象需要手动释放? 在和 `property` 结合的时候怎样有效的避免内存泄露?

- 1> 只要调用了 `alloc`、`copy`、`new` 方法产生了一个新对象, 都必须在最后调用一次 `release` 或者 `autorelease`
- 2> 只要调用了 `retain`, 都必须在最后调用一次 `release` 或者 `autorelease`
- 3> `@property` 如果用了 `copy` 或者 `retain`, 就需要对不再使用的属性做一次 `release` 操作
- 4> 如果用了 `ARC`, 另外讨论

9. 看下面的程序,三次 `NSLog` 会输出什么? 为什么?

```
NSMutableArray* ary = [[NSMutableArray array] retain];
NSString *str = [NSString stringWithFormat:@"test"]; // 1
[strretain]; // 2
[aryaddObject:str]; // 3
NSLog(@"%d",[str retainCount]);
```

```
[strretain]; // 4
[strrelease]; // 3
[strrelease]; // 2
NSLog(@"%d",[str retainCount]);
[aryremoveAllObjects]; // 1

NSLog(@"%d",[str retainCount]);
```

结果：3、2、1

10. OC 中创建线程的方法是什么？如果指定在主线程中执行代码？如何延时执行代码？

1> 创建线程的方法

- NSThread
- NSOperationQueue 和 NSOperation
- GCD

2> 主线程中执行代码

- [self performSelectorOnMainThread: withObject: waitUntilDone:];
- [self performSelector: onThread:[NSThread mainThread] withObject: waitUntilDone:];
- dispatch_async(dispatch_get_main_queue(), ^{
});

3> 延时执行

- double delayInSeconds = 2.0;
dispatch_time_t popTime = dispatch_time(DISPATCH_TIME_NOW,
(int64_t)(delayInSeconds * NSEC_PER_SEC));
dispatch_after(popTime, dispatch_get_main_queue(), ^(void){
});
- [self performSelector: withObject: afterDelay:];
- [NSTimer scheduledTimerWithTimeInterval: target: selector: userInfo: repeats:];