

内存管理

1. 内存管理的作用;
2. 为什么要内存管理, 我们需要注意的问题是什么
3. 内存管理的原理;
4. 什么是黄金法则
5. 在内存管理中 setter 函数的写法
6. @property 参数 (retain) 要注意的问题
7. 什么时候 autorelease ? 与 release 的区别
8. 什么是自动释放池;
9. 阐述内存管理;
10. autorelease 可以使对象计数器减 1 吗?
11. 一个工程中只能有一个自动释放池 (对错)
12. 在手动内存管理中, 尽量都使用 autorelease (对错)

看程序找茬(描述有错则指出并改正)

一. 已知有一个 Book 类:

```
////////////////////////////////////
@interface Book : NSObject
{
    int _value;
}
- (void)setValue:(int)value;
- (void)showValue;
@end
@implementation Book
- (void)setValue:(int)value{
    value = _value;
}
- (void)showValue{
    NSLog(@"value = %d", _value);
}
@end
////////////////////////////////////
```

有问题吗? 有就找出问题

在 main 函数中实现以下程序代码

1.

```
Book * book = [[Book alloc] init];
```

2.


```

Book * book = [[Book alloc] init];
Book * bookTest = book;
[book release];
[bookTest showValue];

```
3.


```

Book * book = [[Book alloc] init];
Book * bookTest = [book retain];
[book release];
[bookTest showValue];

```
4.


```

Book * book = [[Book alloc] init];
Book * bookTest = [book retain];
[book release];
[bookTest showValue];
[book release];

```
5.


```

[[Book alloc] init].value = 100;

```
6.


```

Book * book = [[Book alloc] init];
Book * bookTest = book;
[book release];
[bookTest release];

```
7.


```

Book * book = [[Book alloc] init];
[book release];
book = nil;
Book * bookTest = [book retain];
[bookTest showValue];
[bookTest release];

```
8.


```

[nil release];

```
9.


```

NSMutableString * str1 = [NSMutableString stringWithString:
@"I love iphone"];
NSMutableString*str2 =
[NSMutableString alloc] initWithString: @"I love iphone"];

NSMutableString * str3 = [str1 retain];
[str3 release];
[str2 release];

```

```
[str1 release];
```

```
/*****/
```

二. 有一个 Person 类

下面的类是否有问题

```
@interface Person : NSObject
{
    NSString * _name;
}
@property (nonatomic,copy) NSString* name;
@end
```

```
@implementation Person
@synthesize name = _name;
@end
```

```
/*****/
```

三. 已知有两个类 Book 和 Student

下面类的设计是否有问题

```
@interface Book : NSObject
{
    int _value;
}
- (void)setValue:(int)value;
- (void)showValue;
@end
@implementation Book
- (void)setValue:(int)value{
    value = _value;
}
- (void)showValue{
    NSLog(@"value = %d",_value);
}
@end

#import "Book.h"
@interface Student : NSObject
{
    Book * _book;
    NSString * _name;
    NSMutableArray * _booksArray;
}
}
}
```

```

- (void)setBook:(Book *)book;
- (id)init;
@property (nonatomic,copy)NSString * name;

@end
@implementation Student
@synthesize name = _name;

- (id)init{
    if (self = [super init]) {
        _booksArray = [[NSMutableArray alloc]init];
    }
    return self;
}
- (void)setBook:(Book *)book{
    _book = [book retain];
}
- (void)dealloc {
    [_name release];
    [_book release];
    [super dealloc];
}
@end
/*****

```

四. 有一个 Car 类

找出问题

```

@interface Car : NSObject
{
    int _speed;
}
- (Car *)initWithSpeed:(int)speed;
+ (Car*)carWithSpeed:(int)speed;
@end

@implementation Car
- (Car *)initWithSpeed:(int)speed{
    if ( self = [super init]) {
        speed = _speed;
    }
    return self;
}
+ (Car *)carWithSpeed:(int)speed{
    Car * car = [[Car alloc]initWithSpeed:100];
    [car release];
}

```

```

        return car;
    }
@end
/*****

```

五. 有一个 Car 类

纠错

```

@interface Car : NSObject
- (void)show;
@end
@implementation Car
- (void)show{
    NSLog(@"我是大奔");
}
@end

int main (int argc, const char * argv[])
{
    @autoreleasepool {
        Car * car = [[Car alloc] init];

        NSString* str = @"1000phone";
        NSMutableArray * array = [[NSMutableArray
alloc] init];

        [array addObject:car];
        [array addObject:str];
        [car release];
        [str release];

        Car *carTest = [array objectAtIndex:0];
        [carTest show];

        [array release];
    }
    return 0;
}

```

六. 有一个 Car 类

纠错

```

@interface Car : NSObject

```

```

- (void)show;
@end
@implementation Car
- (void)show{
    NSLog(@"我是大奔");
}
@end
1.
int main (int argc, const char * argv[])
{
    Car * car;
    @autoreleasepool {

        car = [[Car alloc]init];
        [car autorelease];
    }
    [car show];

    NSAutoreleasePool * pool = [[NSAutoreleasePool
alloc]init];

    Car *carTest = [[Car alloc]init];
    [carTest autorelease];
    [pool release];
    [carTest show];

    return 0;
}
2.
int main (int argc, const char * argv[])
{
    Car * car;
    @autoreleasepool {

        car = [[Car alloc]init];
        [car autorelease];
        [car show];

        NSAutoreleasePool * pool = [[NSAutoreleasePool
alloc]init];

        Car *carTest = [[Car alloc]init];

```

```

        [carTest autorelease];
        [carTest show];
        [pool release];
    }
    return 0;
}

3.
int main (int argc, const char * argv[])
{
    Car * car;
    @autoreleasepool {

        car = [[Car alloc] init];

        @autoreleasepool {

            [car autorelease];
        }
        [car show];
    }
    return 0;
}

```

七. 有一个 MyLabel 类

纠错

```

@interface Mylabel : NSObject
{
    NSMutableArray * array;
}
- (void)show;
@end
@implementation Mylabel

- (void)func{
    array = [NSMutableArray array];
    [array addObject:@"1000phone"];
    [array addObject:@"iOS"];
}

- (void)useArray{
    @autoreleasepool {

```

```

        [self func];
    }
}
- (void)show{
    [self useArray];
    NSLog(@"array = %@",array);
}
@end
#import "Mylabel.h"
int main (int argc, const char * argv[])
{
    @autoreleasepool {
        Mylabel * label = [[Mylabel alloc]init];
        [label show];

    }
    return 0;
}

```

八. 有一个 MyLabel 类

纠错

```

@interface Mylabel : NSObject
{
    NSMutableArray * array;
}
- (void)show;
@end
@implementation Mylabel

- (void)func{
    array = [[[NSMutableArray alloc]init]autorelease];
    [array addObject:@"1000phone"];
    [array addObject:@"iOS"];
}

- (void)useArray{
    @autoreleasepool {
        [self func];
    }
}
- (void)show{
    [self useArray];
    NSLog(@"array = %@",array);
}
@end

```



```
#import "Mylabel.h"
int main (int argc, const char * argv[])
{
    @autoreleasepool {
        Mylabel * label = [[Mylabel alloc]init];
        [label show];

    }
    return 0;
}
```