简答题:

1.请写出ARC 的中英文全称

答:automatic release count。

2.内存管理的原则是什么?

答:有加必有减。

3.什么时候对象内存计数器会增加？

答:创建时。被引用时。手动retain时。

4.创建一个对象需要经过那三个步骤？

答:alloc,init，传递给指针。

5. .什么是野指针?什么是僵尸对象?什么是空指针?

答： 野指针：指向对象被销毁。指针指向空白内存的指针。

僵尸对象：retaincount已经归0的对象。

空指针：将nil值赋值给指针后的指针。

6. Set方法的作用是什么？Set方法的好处？

答： 为了避免了直接访问对象成员变量（@pubilc）的安全隐患，修改对象成员变量的赋值时使用有权限访问对象内部内容的set方法。

并且，使用set方法对于变量之间的retaincount可以的掌控性更好。

7. Get方法的作用是什么？

为了避免了直接访问对象成员变量（@pubilc）的安全隐患，读取对象成员变量的赋值时使用有权限访问对象内部内容的get方法。

8. 结构体当中能定义oc对象吗？

答：ARC当中不能。MRC中可以。

9.什么是构造方法，使用构造方法有什么注意点。

答:创建OC对象的方法。 注意与父类之间的关系。

10.请说明NSSet、NSArray、NSDictionary的区别?

答: NSSet：无序。内部不存在key。运行效率高。

NSArray：有序。内部key名称可重复。

NSDictionary：有序。内部key名称可重复。

11.请说明 #include、#import @class各自的作用

答：#include：编译时直接粘贴引用代码。

#import ：编译时粘贴引用代码。且允许有重复的声明。

@class：表示类。

12.点语法本质是什么,写一个点语法的例子,并写上注释

答: 点语法的本质是在编译时转换成setter和getter。

Obj.var = x; //点语法调用了对象Obj内部变量var的setter。

x = Obj.var; //点语法调用了对象Obj内部变量var的getter。

13.@property 有什么作用

答:在新版xcode，@property会自动生成对象的getter和setter。且对于对象内存管理的细节进行定义。

14.调用对象的release 方法会销毁对象吗？

答： 如果一直调用release直到retainCount==0时，会自动调用对象的dealloc方法，即对象被销毁。

15.请说明自动释放池autorelease的作用，和工作机制，它与release有什么不同？

答： 自动释放池autorelease在其所作用的代码块“{}”结束时会对其作用域内部所有对象自动release一次。

Autorelease在代码块最后才会使用一次release。Release可以在任何地方使用。

16.请简述什么是协议？

答: 用以操纵对象和对象之间的异同的规则。遵循协议的变量都会拥有符合协议声明的成员变量和方法。如果没有编译器会警告。

且可以对于规定拥有的成员变量可以区分为“必须的”和“可选的”。

17.id类型是什么，instancetype是什么，有什么区别？

答： id类型是一个可以指向任何NSObject对象的指针类型。

instancetype类型也是一个可以指向任何NSObject对象的指针类型。

两者的区别在于使用instancetype类型可以增加代码的安全性。

18.成员变量名的命名以下划线开头的好处？

答：比如说在setter时，/\* \_var = var \*/易于区分需要赋值的对象和传入的对象。

19.Self的用途是什么？

答：可以指向自己本身的成员变量或方法。

20. Super关键字的作用？

答：可以指向父类的成员变量或方法。

编程题:

1. 查找字符串位置

给NSString 增加个分类,获取字符串http://www.itcast.cn中的itcast开始下标与结束下标的位置（字符串：）

#import <Foundation/Foundation.h>

@interface NSString (discribeSubString)

- (void) getLocation:(NSString \*)fragment;

@end

@implementation NSString (discribeSubString)

- (void) getLocation:(NSString \*)fragment{

if ([self containsString:fragment]) {

NSRange range = [self rangeOfString:fragment];

NSLog(@"%@",[NSString stringWithFormat:@"开始下标是：%zd , 结束下标是：%zd",range.location,(range.location+range.length-1)]);

}

}

@end

int main(int argc, const char \* argv[]) {

@autoreleasepool {

NSString \* a= @"http://www.itcast.cn";

a = @"http://www.itcast.cn";

NSLog(@"%@",a);

NSString \*b = @"itcast";

b = @"itcast";

[a getLocation:b];

}

return 0;

}

2. 设计一个形状类，提供（计算、显示）面积和周长的方法，设计一个长方形类和正方形类 继承形状类，并给长方形类提供长度和宽度成员属性，给正方形类提供边长属性，程序运行 输出长方形和正方形的面积和周长。

#import <Foundation/Foundation.h>

#import <CoreGraphics/CoreGraphics.h>

@interface XXVKShape : NSObject

@property CGFloat circumference;

@property CGFloat area;

@end

@implementation XXVKShape

@end

@interface XXVKSquare : XXVKShape

//{

// CGFloat \_x;

//}

@property CGFloat x;

- (CGFloat)xxcircumference;

- (CGFloat)xxarea;

@end

@implementation XXVKSquare

- (CGFloat)xxcircumference{

self.circumference = \_x\*4;

return self.circumference;

}

- (CGFloat)xxarea{

self.area=\_x\*\_x;

return self.area;

}

@end

@interface XXVKRect : XXVKShape

@property CGFloat a;

@property CGFloat b;

- (CGFloat)xxcircumference;

- (CGFloat)xxarea;

@end

@implementation XXVKRect

- (CGFloat)xxcircumference{

self.circumference = \_a+\_a+\_b+\_b;

return self.circumference;

}

- (CGFloat)xxarea{

self.area=\_a\*\_b;

return self.area;

}

@end

int main(int argc, const char \* argv[]) {

@autoreleasepool {

XXVKRect \*ect=[[XXVKRect alloc]init];

ect.a = 5;

ect.b = 5;

[ect xxarea];

[ect xxcircumference];

NSLog(@"长方形的面积是%f",ect.area);

NSLog(@"长方形的周长是%f",ect.circumference);

XXVKSquare \*are =[[XXVKSquare alloc]init];

are.x = 10;

[are xxarea];

[are xxcircumference];

NSLog(@"正方形的面积是%f",are.area);

NSLog(@"正方形的周长是%f",are.circumference); }

return 0;

}

3. 设计一个日期工具类，要求获取当前日期是星期几

int main(int argc, const char \* argv[]) {

@autoreleasepool {

NSArray \*weekD = @[@"星期日",@"星期1",@"星期2",@"星期3",@"星期4",@"星期5",@"星期6"];

NSDate \*d = [NSDate date];

NSCalendar \*calendar = [NSCalendar currentCalendar];

NSInteger unit = NSCalendarUnitYear|NSCalendarUnitMonth|NSCalendarUnitDay|NSCalendarUnitHour|NSCalendarUnitMinute | NSCalendarUnitSecond|NSCalendarUnitWeekday;

NSDateComponents \*cmp = [calendar components:unit fromDate:d];

NSLog(@"%@",weekD[cmp.weekday-1]);

}

return 0;

}