简答题:

1.请写出ARC 的中英文全称

答:automatic reference counting

2.内存管理的原则是什么?

答:有加有减。每一个new,retain,copy必须对应一个release,autorelease

3.什么时候对象内存计数器会增加？

答:new,retain,copy

4.创建一个对象需要经过那三个步骤？

答:创建内存空间，初始化赋值，返回对象地址

5. .什么是野指针?什么是僵尸对象?什么是空指针?

答： 野指针：指向一个被释放的内存区域的指针

僵尸对象 野指针指向的对象

空指针：内存地址为nil的指针

6. Set方法的作用是什么？Set方法的好处？

答： 设置成员变量

方便成员变量的赋值，而且不需要暴露成员变量

7. Get方法的作用是什么？

答：获取成员变量的值

方便成员变量值的获取，而且不需要暴露成员变量

8. 结构体当中能定义oc对象吗？

答：不能

9.什么是构造方法，使用构造方法有什么注意点。

答:以init开头的方法初始化成员变量

构造方法形式：

if(self = [super init])

{

/\*\*\*\*\* 初始化成员变量 \*\*\*\*\*/

}

retrue self;

10.请说明NSSet、NSArray、NSDictionary的区别?

答:NSSet中的值是无序的

NSArray中的值是有序的

NSDictionary中的值是无序的，而且Key（键值）是不能重复的。

11.请说明 #include、#import @class各自的作用

答：#include 引用头文件

#import引用头文件，可以自动处理重复包含的问题

@class声明类名，在相互引用的时候必须使用

12.点语法本质是什么,写一个点语法的例子,并写上注释

答: get或者set方法

@interface Person

@property int age;

@end

Person p = [[Person alloc] init]

13.@property 有什么作用

答:方便声明私有成员变量，而且可以自动添加set和get方法

14.调用对象的release 方法会销毁对象吗？

答： 并不一定

release只会把引用记数器的数值减1

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： 在自动释放池结束的时候把自动释放池中所有的对象进行一次release操作。

区别：自动释放池release是当自己将要释放的时候，进行release

release释放是即时释放

16.请简述什么是协议？

答: 一组定义的方法名

17.id类型是什么，instancetype是什么，有什么区别？

答： id类型是任意的类型，instancetype也是的

instancetype只能作为函数的返回类型，而id没有限制

18.成员变量名的命名以下划线开头的好处？

答： 和系统@property生成的变量名的形式一致，方便区分成员变量

19.Self的用途是什么？

答： 方便在类中调用自身的对象方法和成员变量

20. Super关键字的作用？

答：调用父类，方便在重载的时候，增加子类的方法的同时，不需要重写父类的方法

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 "NSString+beginWithEnd.h"

@implementation NSString (beginWithEnd)

-(NSString \*)beginWithEndOfNSString:(NSString \*)str

{

NSRange rang = [self rangeOfString:str];

return [NSString stringWithFormat:@"(%lu,%lu)",rang.location+1,rang.location+rang.length];

}

@end

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。

Shape.h

#import <Foundation/Foundation.h>

@interface Shape : NSObject

-(CGFloat)perimeter;

-(CGFloat)area;

@end

Shape.m

#import "Shape.h"

@implementation Shape

@end

Rectangle.h

#import <Foundation/Foundation.h>

#import "Shape.h"

@interface Rectangle : Shape

@property (nonatomic,assign) float length;

@property (nonatomic,assign) float width;

@end

Rectangle.m

#import "Rectangle.h"

@implementation Rectangle

-(CGFloat)perimeter

{

return 2\*(\_length+\_width);

}

-(CGFloat)area

{

return \_length\*\_width;

}

@end

Square.h

#import <Foundation/Foundation.h>

#import "Shape.h"

@interface Square : Shape

@property (nonatomic,assign) float lenght;

@end

Square.m

#import "Square.h"

@implementation Square

-(CGFloat)perimeter

{

return 4\*\_lenght;

}

-(CGFloat)area

{

return \_lenght\*\_lenght;

}

@end

main.m

#import <Foundation/Foundation.h>

#import "Square.h"

#import "Rectangle.h"

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

@autoreleasepool {

// insert code here...

//NSLog(@"Hello, World!");

Square \*square = [[Square alloc] init];

square.lenght = 10;

NSLog(@"边长为%f正方形的面积%f",square.lenght,[square area]);

Rectangle \*rectangle = [[Rectangle alloc] init];

rectangle.width = 20;

rectangle.length =30;

NSLog(@"长为%f，宽为%f的长方形的面积为%f",rectangle.length,rectangle.width,[rectangle area]);

}

return 0;

}

3. 设计一个日期工具类，要求获取当前日期是星期几

main.m

#import <Foundation/Foundation.h>

void test()

{

NSString \*time1 = @"2014-01-01";

NSDateFormatter \*fmt = [[NSDateFormatter alloc] init];

fmt.dateFormat = @"yyyy-MM-dd";

NSDate \*date1 = [fmt dateFromString:time1];

NSDate \*now = [NSDate date];;

// 1.创建一个日历对象

NSCalendar \*calendar = [NSCalendar currentCalendar];

// 2.比较时间的差距

int unit = NSCalendarUnitDay;

NSDateComponents \*cmps = [calendar components:unit fromDate:date1 toDate:now options:0];

NSLog(@"相差%ld天", cmps.day);

int n = cmps.day%7;

switch (n) {

case 0:

NSLog(@"星期三");

break;

case 1:

NSLog(@"星期四");

break;

case 2:

NSLog(@"星期五");

break;

case 3:

NSLog(@"星期六");

break;

case 4:

NSLog(@"星期天");

break;

case 5:

NSLog(@"星期一");

break;

case 6:

NSLog(@"星期二");

break;

default:

break;

}

}

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

@autoreleasepool {

test();

}

return 0;

}