简答题:

1.请写出ARC 的中英文全称

答:auto reference counting 自动引用计数

2.内存管理的原则是什么?

答:如果对一个对象使用了alloc,[Mutable]copy,retain,那么你就必须使用realease或者atuorelease

3.什么时候对象内存计数器会增加？

答:进行了alloc,[mutable]copy,retain操作

4.创建一个对象需要经过那三个步骤？

答:内存分配，定义初始化，返回指针

5. .什么是野指针?什么是僵尸对象?什么是空指针?

答： 野指针是指向的地址不存在的指针，

僵尸对象是已经被回收的对象

空指针是指向的地址没有内容

6. Set方法的作用是什么？Set方法的好处？

答： 对数据进行封装，增加程序的可维护性

7. Get方法的作用是什么？

赋值读取属性的值

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: 为类的实例对象开辟存储空间，把所有成员变量都置零

给类，发送一条alloc消息，还必须调用init来完成初始化

1、必须要调用父类init

2、必须要检验父类的init方法是否成功初始化

3、返回自己的指针

10.请说明NSSet、NSArray、NSDictionary的区别?

答:NSSet是

NSArray是字符串数组

NSDictionary是

11.请说明 #include、#import @class各自的作用

答：#include是C语言中导入头文件

#import是OC中导入头文件

@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

12.点语法本质是什么,写一个点语法的例子,并写上注释

答: 点语法的本质就是调用getter和setter方法

self.name

13.@property 有什么作用

答:生成setter和getter方法的声明和实现同时它还生成一个私有的成员变量

14.调用对象的release 方法会销毁对象吗？

答： 不会，只是会使对象的引用计数减一

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：延长对象的生命周期，吧对象放到离自己最近的那个自动释放池中，当自动释放池销毁的时候，他会把放在其中所有对象进行一次release操作，它和对象在哪创建的没有关系

16.请简述什么是协议？

答: 对象中的一种约定只有方法声明没有实现一种类型

17.id类型是什么，instancetype是什么，有什么区别？

答：

18.成员变量名的命名以下划线开头的好处？

答：区分成员变量与局部变量的不同

19.Self的用途是什么？

答： self在对象方法中，它是调用这个方法的那个对象

self在类方法中就代表类，他可以调用其他类的方法

如果定义与成员变量同名局部变量，还想 操作它就要用self来访问

20. Super关键字的作用？

答： 调用父类的方法

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 <Foundation/Foundation.h>

#import "NSString+Gather.h"

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

{

NSArray \* arr = [NSString gatherLocation:@"http://www.itcast.cn"];

NSLog(@"开始位置: %@, 结束位置: %@", arr[0], arr[1]);

return 0;

}

#import <Foundation/Foundation.h>

@interface NSString (Gather)

// 查找字符串中某个字符串的开始下标和结束下标

+ (NSArray \*)gatherLocation:(NSString \*)str;

@end

#import "NSString+Gather.h"

@implementation NSString (Gather)

// 查找字符串中某个字符串的开始下标和结束下标

+ (NSArray \*)gatherLocation:(NSString \*)str

{

NSRange range = [str rangeOfString:@"itcast"];

// 开始位置

NSUInteger start = range.location;

// 结束位置

NSUInteger end = start + range.length - 1;

NSNumber \* number1 = [NSNumber numberWithInteger:start];

NSNumber \* number2 = [NSNumber numberWithInteger:end];

NSArray \* array = [NSArray arrayWithObjects:number1,number2, nil];

// 返回字符串中某个字符串的开始下标和结束下标

return array;

}

@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。

#import <Foundation/Foundation.h>

@interface Shape : NSObject

// 计算面积

- (float)calculateAreaWithLength:(float)num1 andWide:(float)num2;

// 计算周长

- (float)calculateCircumferenceWithLength:(float)num1 andWide:(float)num2;

@end

#import "Shape.h"

@implementation Shape

// 计算面积

- (float)calculateAreaWithLength:(float)num1 andWide:(float)num2

{

float area = num1 \* num2;

NSLog(@"面积为: %.2f", area);

return area;

}

// 计算周长

- (float)calculateCircumferenceWithLength:(float)num1 andWide:(float)num2

{

float circumference = (num1 + num2) \* 2;

NSLog(@"周长为: %.2f", circumference);

return circumference;

}

@end

#import "Shape.h"

@interface Rectangle : Shape

@property (nonatomic, assign) float length; // 长度

@property (nonatomic, assign) float wide; // 宽度

@end

#import "Rectangle.h"

@implementation Rectangle

@end

#import "Shape.h"

@interface Square : Shape

@property (nonatomic, assign) float len; // 边长

@end

#import "Square.h"

@implementation Square

@end

3. 设计一个日期工具类，要求获取当前日期是星期几

#import <Foundation/Foundation.h>

#import "NSDate+NowDate.h"

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

@autoreleasepool {

NSDate \*date = [NSDate date];

[date nowDate];

}

return 0;

}

#import <Foundation/Foundation.h>

@interface NSDate (NowDate)

// 获取当前日期是星期几

- (void)nowDate;

@end

#import "NSDate+NowDate.h"

@implementation NSDate (NowDate)

// 获取当前日期是星期几

- (void)nowDate

{

NSCalendar \*calendar = [NSCalendar currentCalendar];

NSDate \* date = [NSDate date];

// 获取日期每一个部分

NSDateComponents \*cmps = [calendar components:NSCalendarUnitWeekday fromDate:date];

switch (cmps.weekday) {

case 1:

NSLog(@"今天星期天");

break;

case 2:

NSLog(@"今天星期一");

break;

case 3:

NSLog(@"今天星期二");

break;

case 4:

NSLog(@"今天星期三");

break;

case 5:

NSLog(@"今天星期四");

break;

case 6:

NSLog(@"今天星期五");

break;

case 7:

NSLog(@"今天星期六");

break;

default:

break;

}

}

@end