

## 概览 Overview

---

- 平台相关 platform-related
  1. C/C++ 语言语法，常用数据结构
  2. Objective-C 语法
  3. iOS api
- 平台无关 platform-independent
  1. 多线程与网络编程
  2. 数据结构与算法
  3. 设计模式

### C/C++ 语言语法，常用数据结构

---

- 什么是指针（云来科技）
- 写出 float 与零值比较的 if 语句

```
#define MIN_FLOAT 0.000001  
if (a < MIN_FLOAT && a > 0 - MIN_FLOAT ) {  
    // ...  
}
```

- 如何引用一个已经定义过的全局变量

extern

- **static** 全局变量与普通变量的区别，局部变量呢，函数？

1. **static** 变量是限定作用域的全局变量

2. C 语言中的 **static** 函数是限定作用域的全局函数

3. C++ 里头的 **static** 函数是相对成员函数而言，其调用主体是类，而不是对象

- 关键字 **volatile** 有什么含义？

防止编译器优化

- 队列和栈的区别

一个先进后出，一个先进先出

- 写一个单链表，要求可以插入数据和删除单个数据

```
struct Node {  
  
    int x;  
  
    struct Node *next;  
  
};  
  
struct Node *head;// ...  
  
void insert(struct Node *node) {  
  
    node->next = head->next;  
  
    head->next = node;  
  
}
```

```
void delete(struct Node *node) {  
  
    struct Node *p = head;  
  
    while (p->next) {  
  
        if (p->next == node) {  
  
            p->next = p->next->next;  
  
            free(node);  
  
            break;  
  
        }  
  
        p = p->next;  
  
    }  
  
}
```

```
struct Node {  
    // ...    struct Node *_next;  
};
```

```
extern struct Node *g_head;    // 头结点不算链表内的元素
```

```
void insert(struct Node *obj) {  
    struct Node *temp = g_head->_next;  
    g_head->_next = obj;  
    obj->_next = temp;  
};
```

```
void remove(struct Node *obj) {  
    struct Node *temp = g_head;
```

```
    while (temp) {  
        if (temp->_next == obj) {  
            temp->_next = obj->_next;  
            free(obj);  
            break;  
        }  
  
        temp = temp->_next;  
    }
```

```
}
```

- 堆和栈有什么区别？
- 请声明一个有 10 个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数

```
int func(int x);
```

```
int (*pFunc)(int x);
```

```
pFunc = func;
```

```
int (*aryFunc[10])(int x);
```

```
int *ary[10];
```

```
int (*ary)[10];
```

- 交换两个变量的值，不使用第三个变量。即  $a=3$ ,  $b=5$ ，交换之后  $a=5$ ,  $b=3$ ，请写出两种解法。

```
a = a ^ b; b = a ^ b; a = a ^ b;
```

$a = a + b; b = a - b; a = a - b;$

- 写出标准宏 MIN，这个宏输入两个参数并返回较小的一个

```
#define MIN(a, b) ((a) > (b) ? (b) : (a))
```

## Objective-C 语法

---

- `__block` 的概念

修饰局部变量，这个变量在 block 内外属于同一个地址上的变量，可以被 block 内部的代码修改

- 委托代理与通知中心的区别

1. 一对一和一对 N

2. 耦合性高低

- 实现函数回调的几种方式

1. 委托代理

2. block

3. NotificationCenter

4. performSelector...

5. C 语言的函数指针 `void (*func)();`

- import 与 include 之间的区别



## 避免重复包含

- NSAutoReleasePool 的实现

```
@autoreleasepool {  
  
}
```

```
NSAutoReleasePool *pool =  
[[NSAutoReleasePool alloc] init];
```

```
NSAutoReleasePool *pool2 =  
[[NSAutoReleasePool alloc] init];
```

```
// ...
```

```
[pool release];
```

```
[pool2 release];
```

```
@interface NSAutoReleasePool {
```

```
    NSMutableArray *_arrayObjects;
```

```
}
```

```
@end
```

@implementation NSAutoReleasePool

- (void)dealloc {

[\_arrayObjects release];

}

@end

- (id)autorelease {

[[pool arrayObjects] addObject:self];

[self release];

}

- ARC(automatic reference counting)与非ARC(MRC manual reference counting)之间的区别

1. ARC 不需要书写 release、retain、autorelease

2. \_\_weak、\_\_strong、\_\_bridge、\_\_autoreleasing

- 如何实现 ARC 与非 ARC 混合编译

1. -fno-objc-arc 在 ARC 的环境下，制定某些源文件以非 ARC 的方式编译

2. -fobjc-arc

- ARC 的实现原理

在程序预编译阶段，将 ARC 的代码转换为非 ARC 的代码，自动加入 release、autorelease、retain

- 强引用与弱引用

- 什么是强引用，什么是弱引用？
- 强引用与弱引用哪个更好用？
- 强引用与弱引用的应用场景

1. 大部分的情况下用的都是强引用

## 2. 需要避免相互强引用的时候使用弱引用

- 深拷贝与浅拷贝
- 深拷贝与浅拷贝的应用场景

1. 大部分的情况下用的都是浅拷贝
2. 特殊的业务需求使用深拷贝

- Objective-C 里用到的集合类是什么

NSSet & NSMutableSet

- Objective-C 中有无多继承，如何实现

1. protocol
2. Extension

- 用 Objective-C 写两个线程，对同一个变量，分别进行+/-操作

@synchronized / atomic

- Extension、Category、继承三者之间的关系和区别

1. 三者都是对某个类进行扩展或改写
2. 绝大部分场景使用继承
3. Category 的特点是不需要修改类名，即可对一个类进行扩展或改写

4. Extension 是匿名 Category, 可以追加成员变量

- Objective-C 中如何创建一个线程, 要求可以一直工作, 不会只执行一次就结束

1. break;

2. [thread cancel];

- Why we call Objective-C is runtime language? What does it means?

- 使用 block 和使用 delegate 完成委托模式有什么优点?
- 使用 GCD 以及 block 时要注意些什么? 它们俩是一回事么? block 在 ARC 中和传统的 MRC 中的行为和用法有没有什么区别, 需要注意些什么?
- Objective-C 有私有方法吗, 私有变量呢?