一.语法篇

共有147题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. @property的作用是申明属性及其特性？[判断题]

A. 正确

B. 错误

2. @synthesize的作用是自动生成属性的访问器(getter/setter)方法？[判断题]

A. 正确

B. 错误

3. 代理（delegate）的作用是让一个对象实现另外一个对象的方法?[判断题]

A. 正确

B. 错误

4. #import和#include的区别是：#import可以避免文件被多次引用？[判断题]

A. 正确

B. 错误

5. 在Objective-C中是否支持运算符重载？[判断题]

A. 是

B. 否

6. 在Objective-C中能否在头文件里申明私有方法？[判断题]

A. 可以

B. 不可以

7. Block不允许修改外部变量的值？[判断题]

A. 正确

B. 错误

8. Swift布尔类型的值只能是true或者false？[判断题]

A. 正确

B. 错误

9. Swift语言可以像C语言那样，用非0表示true，用0表示false？[判断题]

A. 正确

B. 错误

10. Swift语言可以像Objective-C那样，使用YES表示true，使用NO表示false？[判断题]

A. 正确

B. 错误

11. Swift的continue语句，用来告诉用户一个循环体停止本次的循环，并立即进入下次的循环。[判断题]

A. 正确

B. 错误

12. 在Swift环境中，无法为函数的参数设置默认值。[判断题]

A. 正确

B. 错误

13. Swift语言支持将函数作为另一个函数的返回类型？[判断题]

A. 正确

B. 错误

14. Swift语言支持将函数作为另一个函数的参数？[判断题]

A. 正确

B. 错误

15. Swift语言中的函数无法同时返回多个值？[判断题]

A. 正确

B. 错误

16. 在Swift环境中，可以为类、结构体和枚举等类型定义静态方法?[判断题]

A. 正确

B. 错误

17. 在为结构体和枚举定义静态方法时，可以使用static关键字?[判断题]

A. 正确

B. 错误

18. 在为类定义静态方法时，可以使用class关键字?[判断题]

A. 正确

B. 错误

19. Swift和Objective-C一样是通过自动引用计数（ARC）技术，来实现对实例的内存管理的？[判断题]

A. 正确

B. 错误

20. 在Swift环境中，类的继承不同于结构体和枚举类型，继承是类所特有的？[判断题]

A. 正确

B. 错误

21. Swift的延迟加载属性是指当第一次被调用的时候，才会对其进行初始化操作的属性？[判断题]

A. 正确

B. 错误

22. 在Swift环境中，元组、枚举和结构属于引用类型，而类则属于值类型？[判断题]

A. 正确

B. 错误

23. 类型检查在Swift中，是使用is和as操作符来实现的？[判断题]

A. 正确

B. 错误

24. Swift语言中的AnyObject，可以代表任何class类型的实例？[判断题]

A. 正确

B. 错误

25. Swift语言中的Any可以表示任何类型，除了方法类型（function types）？[判断题]

A. 正确

B. 错误

26. Swift中的扩展功能（Extension），可以在没有权限获取原始源代码的情况下，扩展某个类型的功能？[判断题]

A. 正确

B. 错误

27. 现有的类、结构或枚举类型的方法，都可以在扩展的帮助下进行添加？[判断题]

A. 正确

B. 错误

28. 在Swift语言中，类型的功能可以使用扩展加入，但重写的功能不能使用扩展？[判断题]

A. 正确

B. 错误

29. 类，结构体和枚举类型都可以遵循协议，并提供具体实现来完成协议定义的方法和功能？[判断题]

A. 正确

B. 错误

30. Swift没有Objective-C中的#define、__OPTIMIZE__宏定义方法？[判断题]

A. 正确

B. 错误

31. Objective-C中的头文件和实现文件，请问.h文件类似C++中的.h文件，.m文件类似.cpp文件？[判断题]

A. 正确

B. 错误

32. Objective-C语言可以和C、C++混合编程吗？[判断题]

A. 可以

B. 不可以

33. Objective-C只有单一继承，没有多重继承？[判断题]

A. 正确

B. 错误

34. 如果Category和原始类中的方法名称冲突，则Category将覆盖原始类的方法，因为Category具有更高的优先级？[判断题]

A. 正确

B. 错误

35. 无法向Category中添加实例变量吗？[判断题]

A. 正确

B. 错误

36. Swift 中枚举可以有初始化方法、也可以添加函数?[判断题]

A. 正确

B. 错误

37. Swift 中的结构体可以添加函数、扩展，和实现 protocol吗[判断题]

A. 可以

B. 不可以

38. Swift 中字符串、数组和字典是以结构体的形式实现的，还是以类的形式实现的？[判断题]

A. 类

B. 结构体

39. Swift 中的 Array、Dictionay、String 都是结构体，传递时会进行拷贝，而不会改变原有的值吗？[判断题]

A. 是的

B. 不是

40. 使用runtime Associate方法关联的对象，需要在主对象dealloc的时候释放么？[判断题]

A. 需要

B. 不需要

41. 可以向编译后得到的类中增加实例变量？[判断题]

A. 可以

B. 不可以

42. 可以向运行时创建的类中添加实例变量？[判断题]

A. 可以

B. 不可以

43. 闭包是值类型还是引用类型的？[判断题]

A. 值类型

B. 引用类型

44. weak属性需要在dealloc中置为nil吗？[判断题]

A. 需要

B. 不需要

45. 在Objective-C中，向nil对象发送消息会导致crash吗？[判断题]

A. 会

B. 不会

46. Objective-C的load和initialize方法只会被调用一次吗？[判断题]

A. 是

B. 不是

47. 调用一个类的静态方法，需要release吗？[判断题]

A. 需要

B. 不需要

48. 与alloc相对应的方法是：[单选题]

A. alloc

B. dealloc

C. realease

D. autorelease

49. 与retain对应的方法是：[单选题]

A. alloc
B. dealloc
C. realease/autorelease
D. copy

50. 一个对象的dealloc方法在什么时被调用？[单选题]

A. 引用计数(retainCount)降为-1
B. 引用计数(retainCount)降为1
C. 引用计数(retainCount)降为2
D. 引用计数(retainCount)降为0

51. 分类（categories）能够向一个已有的类中添加：[单选题]

A. 方法
B. 属性
C. 动作
D. 协议

52. 协议（protocal）可以分为两种，其中以哪种关键字申明的协议可以不被实现：[单选题]

A. @optional
B. @required
C. @none
D. @useful

53. 哪个类是大多数Objective-C类继承的根类，它没有父类 [单选题]

A. NSNull
B. NSInteger
C. NSString
D. NSObject

54. -（void）setName:(NSString *)newName{ self.name = newName; } 被调用的结果是：[单选题]

A. 设置name属性的值
B. 程序死循环
C. 获得name属性的值
D. 复制name属性的值

55. 关键字nil在Objective-C中表示：[单选题]

A. 一个空对象
B. 一个空指针
C. 一个空字符
D. 一个空字节

56. 在Objective-C中，类的成员变量默认被申明为：[单选题]

A. @private
B. @protected
C. @public
D. @package

57. 关于Objective-C++中的异常处理，下面说法最正确的是：[单选题]

A. Objective-C不支持异常处理
B. 在Objective-C++中，Objective-C的异常处理能够捕获C++的异常
C. 在Objective-C++中，Objective-C的异常处理不能捕获到C++的异常
D. 在Objective-C++中，Objective-C和C++的异常处理可以相互捕获异常

58. 在对象的dealloc方法中，关于[super dealloc]；语句说法正确的是：[单选题]

A. [super dealloc];有没有都可以
B. [super dealloc];应该放在dealloc方法内的第一行
C. [super dealloc];应该放在dealloc方法内的最后一行
D. [super dealloc];位置无所谓，只要在dealloc方法里有就可以

59. 下面关于方法：[[[object method1] method2] method3:[object method4]];中的method调用顺序是：[单选题]

A. 1 2 3 4
B. 1 2 4 3
C. 4 1 2 3
D. 4 3 2 1

60. ARC（Automatic Reference Counting）模式下，下面哪个方法是未被禁用的：[单选题]

A. retain
B. copy
C. release
D. autorelease

61. 下列对象中不能调用initWithContentsOfFile方法进行构造的是：[单选题]

A. NSString
B. NSXMLParser
C. NSDictionary
D. NSData

62. OC中包含一个自定义的头文件应该使用：[单选题]

A. #include ""
B. #include<>
C. #import ""
D. #import<>

63. 必须实现的协议方法使用哪个关键字声明：[单选题]

A. @property
B. @interface
C. @required
D. @protocol

64. 声明一个方法：提供Student对象数组按照学号升序排列的判断条件[单选题]

A. -(NSComparisonResult)compareWithNumberForAscendSort:(Student*)otherStudent
B. -(NSInteger)compareWithNumberForAscendSort:(Student*)otherStudent
C. - (BOOL))compareWithNumberForAscendSort:(Student*)otherStudent
D. +(int)compareWithNumberForAscendSort:(Student*)otherStudent

65. OC语言中类的实例变量如果只可以被本类和其子类访问，需要下列哪个修饰符：[单选题]

A. @public
B. @private
C. @protected
D. @friendly

66. 关于集合描述不正确的是：[单选题]

A. 所有的集合都可以添加元素。
B. 集合分为可变集合.不可变集合。
C. 数组，字典和NSSet这三种集合存储数据的方式不同。
D. 所有集合都能保存id类型数据。

67. NSDictionary *dict = [NSDictionarydictionaryWithObject:@"a value" forKey:@"aKey"]; NSLog(@"%@", [dictobjectForKey:@"aKey"]); [dict release];以上内容输出：[单选题]

A. a value
B. aKey : a value
C. 崩溃信息
D. a value : aKey

68. 向OC控制台输出一个字符串应该用：[单选题]

A. NSLog()
B. Printf()
C. NSPrintf()
D. scanf()

69. 名为print::的方法有几个传入参数：[单选题]

B. 1
C. 2
D. 3

70. 下列哪个选项与@synthesize配合使用：[单选题]

A. @end
B. @interface
C. @implementation
D. @property

71. 在Objective-C中，所有类的根类是：[单选题]

A. NSObject
B. NSNull
C. NSCoding
D. UIView

72. 能准确删除“张三”的代码是: NSMutableDictionary * peoples=[NSMutableDictionary dictionaryWithObjectsAndKeys:@"张三",@"左护法",@"李四",@"右使",@"唐sir",@"老大哥", nil]; [单选题]

A. [peoples removeObjectForKey:@"左护法"];
B. [peoples removeObject:@"张三"];
C. [peoples removeObjectAtIndex:0];
D. [peoples removeAllObjects];

73. 使用OC语言输出“This is a Objective-C program.”正确的是： [单选题]

A. NSLog(@"%s",This is a Objective-C program.);
B. NSLog(@"%@",This is a Objective-C program.);
C. NSLog(@"This is a Objective-C program.",%@);
D. D.NSLog(@"This is a Objective-C program.");

74. 协议与委托的描述不正确的是：[单选题]

A. 委托是iOS的一种设计模式
B. 协议里的方法默认是必须实现的
C. 类接受了多个协议，这些协议写在父类后面的<>内，多个协议之间用“.”隔开
D. 定义协议使用@protocol关键字

75. 关于协议描述不正确的是：[单选题]

A. 协议分正式和非正式协议。
B. 在要求确认协议的类必须实现协议中的方法时，可以使用@required。
C. 一个协议可以被多个类确认。
D. 协议中声明的方法，默认是@optional的。

76. 下面关于类概念的描述中,错误的是：[单选题]

A. 类是抽象数据类型的实现
B. 类是具有共同行为的若干对象的统一描述体
C. 类是创建对象的模板
D. 类就是C语言中的结构体类型

77. 在Objective-C中，下面哪些选项是属性的正确声明方式：[单选题]

A. @property(nonatomic,retain)NSStringmyString;
B. @property(nonatomic,assign)NSString * myString;
C. @property(nonatomic,assign)int mynumber;
D. @property(nonatomic,retain)int mynumber;

78. NSString *name = [[NSString alloc] initWithString:@"张三"]; NSLog(@"%d",[name retainCount]);以上代码输出的内容是：[单选题]

A. -1
B. 0
C. 1
D. 2

79. 不会立刻使引用计数器改变的是：[单选题]

A. release
B. alloc
C. autorelease
D. retain

80. OC中的委托最好声明成：[单选题]

A. retain
B. assign
C. copy
D. readonly

81. 经过下列哪种操作后需要使用release方法：[单选题]

A. delegate
B. assign
C. retain
D. dealloc

82. 在OC中类的接口声明关键字为：[单选题]

A. @import
B. @interface
C. @implemention
D. @protocol

83. -(id)setName:(NSString *)name :(NSString *)sex age:(int)age的方法名是：[单选题]

A. -(id)setName:(NSString *)name :(NSString *)sex age:(int)age
B. -setName:name :sex age:age
C. setName: : age:
D. setName:name:age:

84. 将@"2C青年"替换成@"有为青年"的代码是：[单选题]

A. [@"2C青年" stringByAppendingOfString:@"2B" withString:@"有为"];
B. NSRange range = {0,2}; [@"2C青年" stringByReplacingCharactersInRange:rangewithString:@"有为"];
C. NSRange range = {1,2};[@"2C青年" stringByReplacingCharactersInRange:rangewithString:@"有为"];
D. [@"2C青年" stringByReplacingOccurrencesOfString:@"有为" withString:@"2C"];

85. 下列回调机制的理解不正确的是：[单选题]

A. 目标动作对：当两个对象之间有比较紧密的关系时，如视图控制器与其下的某个视图。
B. 代理：也叫委托，当某个对象收到多个事件，并要求同一个对象来处理所有事件时。委托机制必须依赖于某个协议定义的方法来发送消息。
C. 通告机制：当需要多个对象或两个无关对象处理同一个事件时。
D. Block：适用于回调只发生一次的简单任务。

86. 在Xcode中，需要编译混合Objective-C和C++的源码文件，需要将文件格式的后缀改为：[单选题]

A. .c
B. .cpp
C. .mm
D. .m

87. Objective-C声明一个类所要用到的编译指令是：[单选题]

A. @interface SomeClass
B. @protocol SomeClass
C. @implementation SomeClass
D. @autorelease SomeClass

88. 下面哪个方法不属于NSObject的内省（Introspection）方法：[单选题]

A. init
B. isKindOfClass
C. responseToSelector
D. isMemberOfClass

89. 使用protocol时，声明一组可选择实现与否的函数，需要在声明的前一行加上：[单选题]

A. @required
B. @optional
C. @interface
D. @protocol

90. 下面关于#import和#include的描述正确的是：[单选题]

A. #import 是 #include 的替代指令，防止重复引用
B. #import 和 #include 不可以混合使用
C. #import 只用于引用 Objective-C的文件， #include 只用于引用C和C++的文件
D. #import 和 #include 的使用效果完全相同

91. 下面哪个方法可以比较两个NSString *str1, *str2 的异同：[单选题]

A. if(str1 = str2) xxx ;
B. if([str1 isEqualToString:str2]) xxx ;
C. if(str1 && str2) xxx ;
D. if([str1 length] == [str2 length]) xxx;

92. Objective-C有私有方法吗？有私有变量吗？[单选题]

A. 有私有方法和私有变量
B. 没有私有方法也没有私有变量
C. 没有私有方法，有私有变量
D. 有私有方法，没有私有变量

93. 一个类的delegate（代理）的作用不正确的是：[单选题]

A. delegate中的函数在其他类中实现
B. 主要用于不同类型的对象之间一对一传递消息
C. 没有指派则不会触发
D. 可以一个对象的delegate指派给多个其他类型的对象

94. 下面说法不正确的是：[单选题]

A. readwrite 是可读可写特性；需要生成getter方法和setter方法时
B. readonly 是只读特性 只会生成getter方法 不会生成setter方法 ;不希望属性在类外改变
C. assign 是赋值特性，和weak一样，可用weak代替；
D. retain 表示持有特性，setter方法将传入参数先保留，再赋值，传入参数的retaincount会+1;

95. 对于语句NSString*obj = [[NSData alloc] init]; obj在编译时和运行时，分别是什么类型的对象？[单选题]

A. 编译时是NSString的类型；运行时是NSData类型的对象
B. 编译时是NSObject的类型；运行时是NSData类型的对象
C. 编译时是NSData的类型；运行时是NSString类型的对象
D. 编译时是NSObject的类型；运行时是NSString类型的对象

96. 以下关于类别的作用，说法不正确的是：[单选题]

A. 将类的实现分散到多个不同文件或多个不同框架中。
B. 创建对私有方法的前向引用。
C. 向对象添加非正式协议。 继承可以增加，修改或者删除方法。
D. 不能添加属性

97. 下面关于Objective-C中的block定义正确的是： [单选题]

A. tyoedef void(^SuccessBlock)(BOOLsuccess);
B. tyoedef void(^SuccessBlock)(NSStringvalue,BOOL success);
C. tyoedef void^(SuccessBlock)(NSStringvalue,BOOL success);
D. tyoedef void^(SuccessBlock)(NSString*value);

98. 下列关于iOS开发中类方法的使用描述,错误的是:[单选题]

A. 类方法可以调用类方法
B. 类方法不可以调用实例方法,但是类方法可以通过创建对象来访问实例方法
C. 类方法不可以使用实例变量,包括self(可以使用self)
D. 类方法作为消息,可以被发送到类或者对象里面去

99. 以下哪个语句，可以允许Swift代码在执行完一个case分支后，跳转到下一个case分支中。[单选题]

A. continue
B. break
C. fallthrough
D. goto

100. 以下关于Swift语言中的数组，描述错误的是： [单选题]

A. 组成数组的各个变量称为数组的分量，也称为数组的元素，有时也被称为下标变量。
B. 在Swift中，数据值在被存储进入某个数组之前，数据的类型不需要明确。
C. 数组中元素的访问，是通过下标语法实现的。
D. 在Swift中，可以使用数组的append()方法，或者直接使用加号+，往已有的数组中添加新的元素。

101. 以下关于Swift语言的枚举类型，描述错误的是： [单选题]

A. 所谓枚举是指将变量的值一一列举出来，并只限于在列举出来的值的范围内进行取值。
B. Swift语言不须给枚举中的每一个成员都提供值。
C. 字符、整数和浮点类型，可以作为枚举成员的数据类型，字符串则无法作为枚举成员的数据类型。
D. 和Objective-C语言中的枚举不同，你可以为Swift的枚举添加方法。

102. 以下关于Swift语言的结构体类型，描述错误的是： [单选题]

A. 可以为结构体定义常量或变量的属性。
B. 和Objective-C语言中的结构体不同，你可以为Swift的结构体添加方法。
C. 可以给结构体定义下标。
D. 使用structure关键字来进行定义的

103. 如果你用 Swift 写的 class 是继承自 NSObject的话，Swift 会自动为所有的非 private 的类和成员加上哪个标记？ [单选题]

A. @property
B. @objc-no
C. @objc
D. @synchronized

104. Objective-C异常并不是真正的异常，但是当一个异常被抛出到最外层还没被捕获，程序会强行发送哪个信号来中断程序？ [单选题]

A. SIGINT
B. SIGABRT
C. SIGQUIT
D. SIGSEGV

105. 如果管理员发现某个进程终止不了，可尝试发送哪个信号?[单选题]

A. SIGINT
B. SIGABRT
C. SIGKILL
D. SIGSEGV

106. 在哪个Swift版本中，Apple向开发者提供了两个类 JSONEncoder 和 JSONDecoder，使对象和 JSON 之间进行转化变得非常简单？[单选题]

A. Swift 1.0
B. Swift 2.0
C. Swift 3.0
D. Swift 4.0

107. 在Swift4中，以下哪条语句可以获得字符串的长度：[单选题]

A. "xxx".length
B. "xxx".count
C. "xxx".characters.count
D. "xxx".characters.length

108. 单向区间主要分两种：确定上限和确定下限的区间，如6...，表示6及之后的所有数值空间。请问单向区间是由哪个Swift版本带来的？[单选题]

A. Swift 1
B. Swift 2
C. Swift 3
D. Swift 4

109. runtime会将weak对象放入哪种数据结构？[单选题]

A. 栈
B. 队列
C. 堆
D. hash表

110. 拓展（extensions)能够在当前类中增加[多选题]

A. 动作
B. 方法
C. 属性
D. 协议

111. autorelease的作用是：[多选题]

A. 将对象放入自动释放池
B. 将对象的引用计数减一
C. 延后调用release方法
D. 立即调用release方法

112. 在KVC中通用的属性访问器方法是哪两个？[多选题]

A. valueForKey:
B. setValue:forKey:
C. getValeForKey:
D. SetValue:

113. 下面哪些关键字会引起对象的引用计数(retainCount)加一：[多选题]

A. alloc
B. init
C. retain
D. copy

114. 在以.mm为拓展名的文件里，可以包含的代码有哪些？[多选题]

A. C
B. Objective-C
C. C++
D. Swift

115. 下面哪些类中使用到了单例（Singletion）模式？[多选题]

A. FileManager
B. NotificationCenter
C. Application
D. Device

116. 下面各数据类型中，能够直接放进NSArray数组的有：[多选题]

A. BOOL
B. NSInteger
C. NSString
D. NSArray

117. Objective-C的快速枚举对比循环访问的优点，下面说法正确的是：[多选题]

A. 使用快速枚举效率更高
B. 语法更简洁
C. 使用更安全
D. 支持更多的数据类型

118. 以下哪一段代码不会抛出异常：[多选题]

A. NSArray *array = @[1, 2, 3];NSNumber*number = array[3];// @[@1,@ 2,@ 3]
B. NSDictionary *dict = @{@”key”: nil};//value不能为空
C. NSString *str = nil; NSString *str2 =[str substringFromIndex:3];
D. NSString *str = @”hi”;NSString *str2 =[str substringFromIndex:3];

119. 关于Swift语言中的常量和变量，以下说法正确的是：[多选题]

A. 常量所指向的是一个特定类型的值，如数字8或者字符“Swift”。
B. 你可以使用let关键词定义一个常量，使用var关键词定义一个变量。
C. 常量和变量在使用前不需要定义（声明）。
D. 变量的值可以根据需要不断进行修改，而常量的值一经定义，是不能够被修改的。

120. Swift4.0支持哪些输出函数？[多选题]

A. print
B. println
C. debugPrint
D. dump

121. 关于Swift注释，以下说法正确的是：[多选题]

A. 序言性注释的内容主要包括模块的接口、数据的描述和模块的功能；

B. 功能性注释主要包括程序段的功能、语句的功能和数据的状态。

C. 单行注释使用//来表示

D. 多行注释使用/*...*/来表示

122. 关于Swift语言的Double和Float类型，以下说法正确的是：[多选题]

A. 浮点数值是具有小数部分的浮点数，比如3.1415和-0.228。

B. 相比整数来说，浮点数类型能表示更广的数值范围，储存更大或者更小的数字。

C. Double表示64位浮点数，在浮点数需要特别大或者特别精确的时候，可使用此类型。

D. Float表示32位浮点数，在不需要64位精度的时候可使用此类型。

123. 关于Swift字符串中的转义符，以下描述正确的是：[多选题]

A. \\ 表示反斜杠

B. \n 表示换行符

C. \r 表示回车符

D. \t 表示制表符

124. 以下对Swift中的switch条件判断语句，描述正确的是：[多选题]

A. switch语句会尝试把某个值与若干个条件进行匹配。

B. switch语句都由多个case构成，每一个 case 都是代码执行的一条分支。

C. 和Objective-C相同的是，在Swift中，你需要为case分支添加break语句。

D. 当条件判断的情况比较多时，通常用switch语句替换if语句。

125. 以下关于Swift语言中的字典，描述正确的是：[多选题]

A. 字典是一个能够使用类似于键的唯一标识符，来获取值的非序列化集合。

B. 和数组不同的是，字典中的元素并没有具体的顺序，也就是说字典是无序存放元素的。

C. 字典可以存储的键和值，必须提前定义它们的类型。

D. 字典可以存储的键和值，无须提前定义它们的类型。

126. 对Swift中的函数描述正确的语句是：[多选题]

A. Swift使用关键字function进行函数的定义。

B. 函数是一个组织在一起的程序代码集合，用来完成某个特定的功能。

C. 通过函数名以及对应类型的参数值，可调用这个函数。

D. 调用函数时传递的参数的顺序，无需与参数列表中的顺序相同。

127. 关于Swift语言的函数，以下说法正确的是：[多选题]

A. 如果你不确定参数的数量，可通过在变量类型后面加 ...，定义可变参数。

B. 如果你不确定参数的数量，可通过在变量类型后面加 __，定义可变参数。

C. 一个函数最多能有一个可变参数，且必须是函数表中最后的一个。

D. 一个函数可以有多个可变参数，可变参数可以放在参数列表中的任意位置。

128. 关于Swift语言的内置函数，以下描述正确的是：[多选题]

A. abs绝对值函数，用来返回指定参数的正值。当参数为负数时，返回对应的正数；当参数为正数时，结果将保持不变。

B. min最小值函数，用来返回不定数量的参数之间的最小值。

C. max最小值函数，用来返回不定数量的参数之间的最大值。

D. map函数通常用于将数组中的每个元素，通过指定的方法进行转换。

129. 以下有关Swift语言中的类和结构体，描述正确的是：[多选题]

A. 两者都可以定义属性以存储值。
B. 两者都可以定义方法以提供功能。
C. 两者都可以定义下标脚本，用于允许使用下标语法访问值。
D. 两者都可以定义初始化器用于初始化状态。

130. 以下有关Swift语言中的类和结构体，描述正确的是：[多选题]

A. 两者都可以被扩展，以实现默认所没有的功能。
B. 两者都可以遵循协议，以针对特定类型提供标准功能。
C. 类可以继承另一个类，结构体也可以继承另一个结构体。
D. 类和结构体都可以添加析构方法。

131. 在Swift语言中，使用扩展可以完成哪些任务？[多选题]

A. 添加计算型属性和计算静态属性。
B. 定义实例方法和类型方法。
C. 提供新的构造器。
D. 使一个已有类型符合某个接口。

132. 以下关于Swift语言中的可选Optional类型，描述正确的是：[多选题]

A. Optional类型是一个enum枚举类型，包含None和Some两个元素。
B. Optional类型是一个enum枚举类型，包含Blank和Some两个元素。
C. 在通过Some(T)包装wrap原始值之后，在使用Optional的时候，就要进行相应的拆包操作。
D. 在通过Some(T)包装wrap原始值之后，在使用Optional的时候，无需进行相应的拆包操作。

133. 在 Swift 中，我们可以给哪些类型定义subscript下标脚本的语法？[多选题]

A. 类
B. 协议
C. 结构体
D. 枚举

134. 以下关于Swift语言的下标说法正确的是：[多选题]

A. 下标脚本使用 subscript 关键字来定义
B. 下标脚本使用 get、set 来定义读、写属性，并不需要get和set都有，可以只读，并且读必须有。
C. 定义 set 属性时，传入的参数默认名称为 newValue。并且 newValue 的类型和 subscript 函数返回值相同。
D. 下标脚本允许你通过在实例后面的方括号中传入一个或者多个的索引值来对实例进行访问和赋值。

135. Object-C具有哪些动态特性？[多选题]

A. 动态类型
B. 动态名称
C. 动态绑定
D. 动态加载

136. 在Objective-C语言中，除了哪几个方法之外创建的对象都是autorelease的？[多选题]

A. init
B. alloc
C. new
D. copy

137. Objective-C有哪些遍历（迭代）方式？[多选题]

A. 使用for循环
B. 使用NSEnumerator遍历
C. 使用for...In遍历
D. 基于Block的遍历方式

138. Objective-C的Category可以实现哪些功能？[多选题]

A. 将类的实现分散到多个不同文件或不同框架中。
B. 创建对私有方法的前向引用。
C. 向对象添加正式协议。
D. 向对象添加非正式协议。

139. 以下哪些情况会在Objective-C中产生异常？[多选题]

A. NSDictionary删除为nil的key
B. NSDictionary添加nil的对象
C. .h文件里函数名，却忘了修改.m文件里对应的函数名
D. 使用第三方库时，没有添加”-ObjC” flag

140. 以下哪些情况会在Objective-C中产生异常？[多选题]

A. NSDictionary插入nil的对象
B. NSURL的初始化，传入nil的http地址
C. 数组最大下标处理错误
D. 如果一个数组刚刚初始化，还是空的，就对它进行相关操作

141. Mach异常是指最底层的内核级异常，以下哪些属于Mach异常？[多选题]

A. EXC_BAD_ACCESS (Bad Memory Access)
B. EXC_BAD_INSTRUCTION运行了非法的指令，往往是运行指令的参数不对（0或者nil的参数）
C. EXC_RESOURCE程序资源上限（cpu占用过高或者内存不足）。
D. EXC_GUARD一些C函数访问错误导致的异常。

142. 最常见的Mach异常：EXC_BAD_ACCESS，一般是由于哪两种情况造成的？[多选题]

A. 访问非法地址(SIGBUS信号)
B. 访问了尚未实现的方法
C. 访问了被回收掉的内存(SIGSEGV信号)
D. 访问数组时出现了索引越界

143. 在以下列出的信号中，程序不可捕获、阻塞或忽略的信号有哪两个：[多选题]

A. SIGTSTP
B. SIGPWR
C. SIGKILL
D. SIGSTOP

144. 在以下列出的信号中，默认会导致进程退出的信号有：[多选题]

A. SIGALRM
B. SIGHUP
C. SIGKILL
D. SIGTERM

145. 以下哪些是常见的 Crash 场景？[多选题]

A. 访问了僵尸对象
B. 访问野指针
C. 数组越界
D. 访问了不存在的方法

146. autoreleasepool是以一个队列数组的形式实现的，主要通过以下哪三个函数完成：[多选题]

A. objc_autoreleasepoolPush
B. objc_autoreleasepoolPop
C. objc_autoreleasepoolMov
D. objc_autorelease

147. 以下哪些是Runloop可以做到的？[多选题]

A. 使程序一直运行接受用户输入
B. 决定程序在何时应该处理哪些Event
C. 调用解耦
D. 节省CPU时间

1 : A	2 : A	3 : A	4 : A	5 : B	6 : B	7 : A	8 : A
9 : B	10 : B	11 : A	12 : B	13 : A	14 : A	15 : B	16 : A
17 : A	18 : A	19 : A	20 : A	21 : A	22 : B	23 : A	24 : A
25 : A	26 : A	27 : A	28 : A	29 : B	30 : A	31 : A	32 : A
33 : A	34 : A	35 : A	36 : A	37 : A	38 : B	39 : A	40 : B
41 : B	42 : A	43 : B	44 : B	45 : B	46 : A	47 : B	48 : B
49 : C	50 : D	51 : A	52 : A	53 : D	54 : B	55 : A	56 : B
57 : C	58 : C	59 : B	60 : B	61 : B	62 : C	63 : C	64 : A
65 : C	66 : A	67 : C	68 : A	69 : C	70 : D	71 : A	72 : A
73 : D	74 : C	75 : D	76 : D	77 : C	78 : A	79 : C	80 : B
81 : C	82 : B	83 : C	84 : B	85 : B	86 : C	87 : A	88 : A
89 : B	90 : A	91 : B	92 : C	93 : D	94 : C	95 : A	96 : D
97 : A	98 : C	99 : C	100 : B	101 : C	102 : D	103 : C	104 : B
105 : C	106 : D	107 : B	108 : D	109 : D	110 : B,C	111 : A,C	112 : A,B
113 : A,C,D	114 : A,B,C	115 : A,B,C,D	116 : C,D	117 : A,B,C	118 : C,D	119 : A,B,D	120 : A,C,D
121 : A,B,C,D	122 : A,B,C,D	123 : A,B,C,D	124 : A,B,D	125 : A,B,C	126 : B,C	127 : A,C	128 : A,B,C,D
129 : A,B,C,D	130 : A,B	131 : A,B,C,D	132 : A,C	133 : A,C,D	134 : A,B,C,D	135 : A,C,D	136 : B,C,D
137 : A,B,C,D	138 : A,B,D	139 : A,B,C,D	140 : A,B,C,D	141 : A,B,C,D	142 : A,C	143 : C,D	144 : A,B,C,D
145 : A,B,C,D	146 : A,B,D	147 : A,B,C,D					

二.常识篇

共有72题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. NSObject是一个根类，几乎所有的类都是从它派生而来。但是根类并不拥有其它类都有的alloc和init方法？[判断题]

A. 正确

B. 错误

2. UIResponder可以让继承它的类响应移动设备的触摸事件，由于可能有多个对象响应同一个事件，iOS将事件沿响应链向上传递？[判断题]

A. 正确

B. 错误

3. 手机基站定位特点是定位速度在几种定位方式中是最快的，并且耗电最小，只是误差范围比较大？[判断题]

A. 正确

B. 错误

4. WIFI定位的支持范围比基站定位广？[判断题]

A. 正确

B. 错误

5. GPS定位的特点是耗电最多，定位速度也最慢，但是优点是定位的精度最高？[判断题]

A. 正确

B. 错误

6. 加速计只能侦测物体的移动行为，并不具备精确侦测物体角度改变的能力？[判断题]

A. 正确

B. 错误

7. 陀螺仪可以侦测物体水平改变的状态，但无法计算物体移动的激烈程度？[判断题]

A. 正确

B. 错误

8. 本地化的资源文件，在应用程序的Bundle中，会根据语言的不同分离成不同的文件吗？[判断题]

A. 正确

B. 错误

9. 开发者可以将项目中的每一个文本、图标和图形文件等进行本地化处理，但是无法对音频、视频、Storyboard文件进行本地化处理。[判断题]

A. 正确

B. 错误

10. 关于App Store，如果用户按年订购，开发者第一年获得受益的70%，如果第二年继续购买订阅服务，开发者就可以获得85%的收益？[判断题]

A. 正确

B. 错误

11. 当向App Store上传IPA文件时，每次上传IPA包的版本号可以相同，但Build号不能相同？[判断题]

A. 正确

B. 错误

12. 应用审核成功并发布后，您将无法修改应用的分类、名称、截图和关键词?[判断题]

A. 正确

B. 错误

13. 应用审核成功并发布后，您无法修改应用的促销文字、版权信息？[判断题]

A. 正确

B. 错误

14. 如果您需要给苹果审核人员写邮件，可以使用中文书写吗？[判断题]

A. 可以

B. 不可以

15. 下载Xcode尽量别用迅雷，迅雷有可能会提供包含后门的Xcode？[判断题]

A. 正确

B. 错误

16. Core ML的已训练模型 (trained model)是指在对一组训练数据应用了某个机器学习算法后，所生成的一组结果吗?[判断题]

A. 是

B. 不是

17. DateFormatter 这种比较消耗性能的类，如果分配内存次数较多的话，需要做一个静态变量，没必要每次都分配? [判断题]

A. 正确
B. 错误

18. iOS开发者账户中，最多能够添加多少台设备号？[单选题]

A. 1000
B. 100
C. 99
D. 999

19. 最多可以为每个内购项目创建多少个促销代码？[单选题]

A. 1000
B. 100
C. 99
D. 999

20. MVC模式中的M、V、C分别指：[单选题]

A. 模态、视图、控制器
B. 模型、视图、控制器
C. 模型、控制器、视图
D. 视图、模型、控制器

21. UDID是由多少位十六进制字符串组成？[单选题]

A. 20
B. 30
C. 40
D. 50

22. 在iOS开发环境下，后缀为.a的文件又叫做：[单选题]

A. 可执行文件
B. 类文件
C. 头文件
D. 静态库

23. iPhone、iPad、iTouch中使用的架构是：[单选题]

A. arm
B. i386
C. x86
D. IA-32

24. 关于应用程序目录在更新、恢复和更新过程的变化，下面说法错误的是：[单选题]

A. 在备份过程中 /Library/Caches目录会被备份
B. 在应用程序更新过程中，/Documents目录会被保留
C. 在备份过程中/tmp目录将不会被备份
D. 在应用程序更新过程中，/Library/Preferences目录会被保留

25. iPhone上，不能被应用程序直接调用的系统程序是：[单选题]

A. 通讯录
B. 短信
C. 日历
D. 邮件

26. 应用程序大小超过多少时，只能通过WiFi从App Store上下载：[单选题]

A. 5MB
B. 10MB
C. 20MB
D. 100MB

27. 可上传至App Store的App应用最大体积是多少？[单选题]

A. 4GB
B. 3GB
C. 2GB
D. 8GB

28. 面向对象的特性不包括：[单选题]

A. 封装
B. 继承
C. 多态
D. 构造

29. 内存管理理解不正确的是：[单选题]

A. 程序A里有一段内存被成功申请完成之后，内存计数器就从0变为1 (这个过程是alloc);
B. 程序B里要使用已存在内存，那么内存计数器从1变为2 (这个过程是retain或者copy);
C. 紧接着程序A不需要这个内存了，那么程序A就把这个内存计数器减1 (这个过程是release);
D. 当系统发现这个内存计数器变为小于等于0，那么就调用垃圾回收程序把这段内存回收(这个过程是dealloc)；

30. MVC优点不正确的是：[单选题]

A. 低耦合性
B. 高重用性和可适用性
C. 较低的生命周期成本
D. 代码高效率

31. Shell中，将command1的输出作为command2的输入应该使用的命令是：[单选题]

A. command1 && command2
B. command1 > command2
C. command1 & command2
D. command1 | command2

32. 下面那项不是动态语言的特性：[单选题]

A. 在运行时替换一个类
B. 在运行时动态加载lib文件
C. 在运行时修改对象中的方法
D. 在运行时增加对象的方法

33. 下面哪个不属于对象数据序列化方法：[单选题]

A. JSON
B. Property List
C. XML
D. HTTP

34. 下列不属于iOS存储方式的是：[单选题]

A. FileManager
B. 归档
C. SQLite
D. CoreData

35. IP Phone的原理是什么?[单选题]

A. IPV4
B. DHCP
C. IPV6
D. DNS

36. 哪个Xcode版本带来了Wireless development无线调试功能？[单选题]

A. Xcode 6
B. Xcode 7
C. Xcode 8
D. Xcode 9

37. CLLocationManager定位的哪种精准度常用于周边的酒店、机场等类型的应用？[单选题]

A. kCLLocationAccuracyNearestTenMeters
B. kCLLocationAccuracyHundredMeters
C. kCLLocationAccuracyKilometer
D. kCLLocationAccuracyBest

38. 使用哪个工具可以进行地理编码和反地理编码的工作？[单选题]

A. CLLocationManagerDelegate
B. CLLocationManager
C. CLLocation
D. CLGeocoder

39. 在iOS开发中，陀螺仪和加速计的使用是通过哪个框架实现的？[单选题]

A. Core Data
B. Core Image
C. Core Motion
D. Core Location

40. CMMotionManager对象的accelerometerUpdateInterval属性的值为0.1，表示每秒钟更新多少次的加速计的状态? [单选题]

A. 1
B. 10
C. 100
D. 1000

41. 在iTunes Connect管理平台，哪个项目可以提供展示应用程序的下载量、用户活跃度、用户地区分布、设备类型、程序版本类型、Crash等信息？[单选题]

A. 我的App
B. App分析
C. 销售和趋势
D. 用户和职能

42. 描述应用程序的关键词，可以用来帮助客户在App Store中搜索到精确的结果。它的长度限制在多少个字符以内？[单选题]

A. 50
B. 100
C. 200
D. 500

43. NSFileManager 是 用来管理和操作文件、目录等文件系统相关联内容的类。它隶属于哪个框架？[单选题]

A. Foundation
B. UIKit
C. Core Data
D. Core Motion

44. 上传至App Store的IPA文件，最大体积是：[单选题]

A. 1G
B. 2G
C. 4G
D. 8G

45. 如果App1需要跳转到App2，可以使用以下哪种方式？[单选题]

A. URL Scheme
B. Keychain
C. UIPasteboard
D. UIDocumentInteractionController

46. Core ML 的运行需要使用哪种模型格式？[单选题]

A. .model
B. .data
C. .mlmodel
D. .plist

47. 以下哪项苹果推出的技术可以将自然语言的字符串标记为单词、确定词性和词根、划分出人名地名和组织名称、告诉你字符串使用的语言和语系？[单选题]

A. Core Data
B. NSLinguisticTagger
C. SiriKit
D. Core Text

48. 应用程序在后台一般可以运行多久？[单选题]

A. 10秒左右
B. 1分钟左右
C. 10分钟左右
D. 1小时左右

49. 关于XML和JSON，下列说法正确的有：[多选题]

A. XML是一种类似HTML的语言，JSON是一种轻量级的数据交换格式。
B. XML是基于键值对（key/value）的结构
C. JSON有DOM和SAX两种解析方式
D. JSON和XML之间可以相互转换

50. 在XCode上创建一个iOS工程，会自动添加进工程的Frameworks有哪些：[多选题]

A. UIKit.framework
B. Foundation.framework
C. CoreGraphics.framework
D. CoreData.framework

51. 以下哪些Cocoa Touch框架，用于数据管理方法：[多选题]

A. Core Data
B. WebKit
C. Core Location
D. SQLite

52. 当开发者免费加入Apple开发者计划之后，以下说法正确的？[多选题]

A. 无法使用beta版的iOS SDK
B. 可以使用beta版的iOS SDK
C. 无法向App Store发布你的产品
D. 可以向App Store发布你的产品

53. 当开发者加入Apple开发者计划并支持年费之后，以下说法正确的？[多选题]

A. 可以使用beta版的iOS SDK
B. 无法进行真机测试
C. 可以向App Store发布你的产品
D. 无法向App Store发布你的产品

54. 关于加入付费的开发者计划，以下说法正确的是：[多选题]

A. 个人开发者账户，每年99美金。可以在App Store发布产品或在100个iOS设备上进行测试。
B. 公司开发者账户，每年99美金。可以在App Store发布产品或在100个iOS设备上进行测试。申请该类型的账户时，需要填写公司的邓白氏编码（D-U-N-S）。
C. 企业开发者账户，每年299美金。可以在企业内部进行无设备数量限制的分发。
D. 企业开发者账户同样可以在App Store中发布产品。

55. 关于类的继承，以下说法正确的是：[多选题]

A. 继承是在一些通用类的基础上构造、建立和扩充新类的最有效的手段。
B. 继承简化了人们对事物的认识和描述，能清晰体现相关类之间的层次结构关系。
C. 继承提供了软件复用功能，能够减小代码的冗余度，增加程序的重用性。
D. 继承通过增强一致性，来减少模块间的接口和界面，大大增加了程序的易维护性。

56. 关于MVC模式中的M、V、C三者之间的通信方式，以下描述正确的是：[多选题]

A. 数据模型Model通过通知Notification和键值观察KVO机制与控制器Controller间接通信。
B. 控制器通过设置View的DataSource属性，设置视图的数据源，如UITableView。
C. 视图View通过动作Action Target向视图控制器ViewController报告事件的发生，例如用户点击了视图。
D. 视图View通过Delegate委托（代理），向视图控制器报告事件的发生。如UIAlertViewDelegate、UITextFieldDelegate。

57. 以下哪些技术可以实现iOS的数据持久化？[多选题]

A. plist属性列表
B. 对象归档
C. SQLite3
D. Core Data

58. Core Data支持哪些类型的持久化数据存储方式？[多选题]

A. SQLiteStore
B. XMLStore
C. BinaryStore
D. InMemoryStore

59. iOS设备支持哪些定位方式：[多选题]

A. 手机基站定位
B. WIFI定位
C. 北斗卫星定位
D. GPS定位

60. 在iOS设备中，陀螺仪的主要用途有哪些？[多选题]

A. 通过陀螺仪配合GPS设备，手机的导航能力达到前所未有的水准。
B. 和iPhone和iPad上的摄像头配合使用，增强拍摄时的防抖性。
C. 陀螺仪相当于一个立体的鼠标，所以经常被用在赛车、战机类的游戏中，可通过摇晃设备来控制赛车和战机的方向。
D. 微信的摇一摇功能，通过摇晃手机可以匹配到同一时段触发该功能的微信用户

61. 陀螺仪主要包含哪几个组件？[多选题]

A. 陀螺仪帧
B. 放置轴
C. 方向坐标系
D. 转子

62. 蓝牙设备包含哪些工作状态：[多选题]

A. 准备（standby）
B. 广播（advertising）
C. 监听扫描（Scanning）
D. 已连接（Connected）

63. 蓝牙4.0支持哪些iOS版本？[多选题]

A. iOS 4
B. iOS 5
C. iOS 6
D. iOS 7

64. iOS的系统架构可以分为哪些层次？[多选题]

A. 核心操作系统层（Core OS layer）
B. 核心服务层（Core Services layer）
C. 媒体层（Media layer）
D. 可触摸层（Cocoa Touch layer）

65. Core ML机器学习框架支持哪些平台？[多选题]

A. iOS
B. macOS
C. tvOS
D. watchOS

66. Core ML支持哪些机器学习技术？[多选题]

A. 深度神经网络
B. 循环神经网络
C. 卷积神经网络
D. 支持向量机

67. Core ML 支持哪些机器学习模型？[多选题]

A. Neural Network
B. Tree Ensemble
C. Support Vector Machine
D. Generalized Linear Model

68. 以下关于Predicate谓词，描述正确的是：[多选题]

A. 能使用一般谓词的不要使用正则表达式，以提高程序的性能。
B. 尽量不使用嵌套，最好把条件拆分。
C. 字符串常量使用"和""是一样的。
D. 关键字不区分大小写，比如：tip和TIP是一样的。

69. 从在屏幕上点击你的app icon开始，到应用执行到main()方法或者执行到applicationWillFinishLaunching的过程中，共执行了哪些方法？[多选题]

A. dylib loading time 载入动态库。
B. ebase/binding time 重构和绑定。
C. ObjC setup time 在Objective-C的运行时(runtime)，需要对类(class)，类别(category)进行注册。
D. initializer time 这一份指代的是执行+initialize方法的时间。

70. 以下关于代码注释，说法正确的是：[多选题]

A. 最好的代码是不需要注释的 尽量通过合理的命名
B. 良好的代码把含义表达清楚 在必要的地方添加注释
C. 注释需要与代码同步更新
D. 如果做不到命名尽量的见名知意的话，就可以适当的添加一些注释或者mark

71. 以下关于iOS Code Review，描述正确的是：[多选题]

A. 文件行数偏多，可以根据功能拆分为不同的文件，让每个类各司其职。
B. 代码风格统一，比如函数名后面大括号的位置，保持书写一致。
C. 分支 if else 尽量配对使用，保证逻辑完整，if 里面尽量包含主要的处理逻辑。
D. 检查代码中的类名、变量的命名风格是否统一。

72. method swizzling交换方法有哪几种实现方式：[多选题]

A. 利用 class_exchangeImplementations 交换两个方法的实现
B. 利用 class_replaceMethod 替换方法的实现
C. 利用 method_exchangeImplementations 交换两个方法的实现
D. 利用 method_setImplementation 来直接设置某个方法的IMP

1 : B	2 : A	3 : A	4 : B	5 : A	6 : A	7 : A	8 : A
9 : B	10 : A	11 : A	12 : A	13 : B	14 : A	15 : A	16 : A
17 : A	18 : B	19 : B	20 : B	21 : C	22 : D	23 : A	24 : A
25 : C	26 : C	27 : A	28 : D	29 : B	30 : D	31 : D	32 : B
33 : D	34 : A	35 : C	36 : D	37 : B	38 : D	39 : C	40 : B
41 : B	42 : B	43 : A	44 : C	45 : A	46 : C	47 : B	48 : A
49 : A,D	50 : A,B,C	51 : A,D	52 : A,C	53 : A,C	54 : A,B,C	55 : A,B,C,D	56 : A,B,C,D
57 : A,B,C,D	58 : A,B,C,D	59 : A,B,D	60 : A,B,C,D	61 : A,B,C,D	62 : A,B,C,D	63 : C,D	64 : A,B,C,D
65 : A,B,C,D	66 : A,B,C,D	67 : A,B,C,D	68 : A,B,C,D	69 : A,B,C,D	70 : A,B,C,D	71 : A,B,C,D	72 : B,C,D

三.界面篇

共有83题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. UIResponder是UIView的父类？[判断题]

A. 正确

B. 错误

2. UIControl是UIView的父类？[判断题]

A. 正确

B. 错误

3. UIWindow提供一个描画内容的表面，是所有其他视图的根容器，每个应用程序只能拥有一个窗口？[判断题]

A. 正确

B. 错误

4. UIView视图是所有控件的父类。控件用于响应用户的交互，而UIVIew则负责内容的显示和布局？[判断题]

A. 正确

B. 错误

5. UIControl类几乎是所有交互控件的父类，如按钮，滑块、文本框等。所以UIControl类负责根据触摸事件触发相应的动作？[判断题]

A. 正确

B. 错误

6. iOS应用程序启动后，创建的第一个视图控件就是UIWindow？[判断题]

A. 正确

B. 错误

7. 在iOS应用程序中，窗口对象并没有像关闭框或标题栏这样的区域，所以用户不能直接对其进行关闭？[判断题]

A. 正确

B. 错误

8. 在macOS中，NSWindow的父类是NSResponder。在iOS系统中，UIWindow的父类是UIResponder？[判断题]

A. 正确

B. 错误

9. 在macOS中，NSWindow的父类是NSResponder。在iOS系统中，UIWindow的父类是UIView？[判断题]

A. 正确

B. 错误

10. UIView中的alpha属性是一个浮点值，取值范围在0~1.0，表示从完全透明到完全不透明？[判断题]

A. 正确

B. 错误

11. 在默认情况下，当向一个视图中添加一个子视图时，如果子视图的区域超出了父视图的范围，子视图超出的部分仍然会在屏幕上正常显示?[判断题]

A. 正确

B. 错误

12. 在对UIView视图进行放大、缩小或旋转时，center属性的值会发生改变？[判断题]

A. 正确

B. 错误

13. UIButton是UIView的一个子类？[判断题]

A. 正确

B. 错误

14. Target-Action机制可以对同一个控件绑定多个Action？[判断题]

A. 正确

B. 错误

15. 如果你的视图并没有指定tintColor属性，那么这个视图将使用父类中的tintColor属性？[判断题]

A. 正确

B. 错误

16. UIButton中受到tintColor影响的范围仅有title和image的颜色？[判断题]

A. 正确

B. 错误

17. UISwitch的大小设置是无效的？[判断题]

A. 正确

B. 错误

18. UITextField支持代理模式delegate，但是不支持Target-Action的机制？[判断题]

A. 正确

B. 错误

19. 在iOS中只有继承自UIResponder类的对象才能处理触摸事件？[判断题]

A. 正确

B. 错误

20. UIViewController视图控制器、UIApplication都继承自UIResponder？[判断题]

A. 正确

B. 错误

21. 如果UIWindow对象无法处理某个触摸事件，则该触摸事件将被丢弃？[判断题]

A. 正确

B. 错误

22. UITapGestureRecognizer属于非连续类型的手势?[判断题]

A. 正确

B. 错误

23. UIPinchGestureRecognizer属于非连续类型的手势?[判断题]

A. 正确

B. 错误

24. Storyboard事实上是由XML可扩展标记语言构成的文本文件？[判断题]

A. 正确

B. 错误

25. 在LaunchScreen.storyboard故事板中，不能使用 UIWebView，因为这里是不能使用网络的。[判断题]

A. 正确

B. 错误

26. 可以为iPhone和iPad指定不同的LaunchScreen.storyboard故事板？[判断题]

A. 正确

B. 错误

27. LaunchScreen.storyboard故事板可以使用UIView或者UIViewController，而自定义的子类则不可以使用。[判断题]

A. 正确

B. 错误

28. UIKit 只能在 iOS平台中使用，无法在mac OS平台使用?[判断题]

A. 正确

B. 错误

29. QuartzCore框架可以直接使用 UIImage、UIColor？[判断题]

A. 正确

B. 错误

30. CALayer和UIView一样也可以处理用户的触摸事件。[判断题]

A. 正确

B. 错误

31. UIView 可以通过 subviews 属性访问所有的子视图，类似地，CALayer 也可以通过 sublayers 属性访问所有的子层? [判断题]

A. 正确

B. 错误

32. UIView 可以通过 superview 属性访问父视图，类似地，CALayer 也可以通过 superlayer 属性访问父层？[判断题]

A. 正确

B. 错误

33. 如果两个 UIView 是父子关系，那么它们内部的 CALayer 也是父子关系吗？[判断题]

A. 正确
B. 错误

34. 因为 UIView 对象已经是它内部根层的 delegate，所以不能再将某个 UIView 设置为 CALayer 的 delegate？[判断题]

A. 正确
B. 错误

35. 在iOS11上，如果APP在启动时图标的四周出现黑色，是因为图标的四角的圆角，并且周围为透明像素。[判断题]

A. 正确
B. 错误

36. LayoutSubViews在什么时候被调用？[判断题]

A. 当View本身的frame改变时
B. 当View本身的bounds改变时

37. UIViewController的didReceiveMemoryWarning方法在什么时会被调用：[单选题]

A. 外接来电
B. 系统错误
C. 系统内存不足
D. 系统CPU被占用100%

38. 下面哪个类在iPhone应用程序开发时不能使用：[单选题]

A. UITabViewController
B. UINavigationController
C. UISplitViewController
D. UITableViewController

39. 下面不是标准UIButton类型的是：[单选题]

A. UIButtonTypeRoundedRect
B. UIButtonTypeContactAdd
C. UIButtonTypeRadio
D. UIButtonTypeInfoLight

40. 在给UIButton设置图片时，为了让图片适应按钮的大小，需要将按钮的contentMode设置为：[单选题]

A. UIViewContentModeScaleToFill
B. UIViewContentModeScaleAspectFit
C. UIViewContentModeScaleAspectFill
D. UIViewContentModeCenter

41. 关于UIControl和UIView之间的关系，下面说法正确的是：[单选题]

A. UIControl继承于UIView
B. UIView继承于UIControl
C. UIControl与UIView之间没有继承关系
D. UIControl和UIView都是直接继承于UIResponder

42. 下列UIView的方法中，哪一个在iOS 5.0前后的系统调用机制不同：[单选题]

A. addSubView
B. layoutSubView
C. drawRect
D. removeFromSuperView

43. 构造UIImage的方法中，占用内存最大的哪个方法：[单选题]

A. imageNamed
B. imageWithContentsOfFile
C. imageWithData
D. 三者一样大

44. UITableView中cell的复用是由几个数组实现的：[单选题]

A. 1
B. 2
C. 3
D. 4

45. 点击Button响应链中最终得到响应的是：[单选题]

A. Window
B. Application
C. AppDelegate
D. UIViewController

46. 在UIKit中，frame与bounds的区别是：[单选题]

A. frame 是 bounds 的别名
B. frame 是 bounds 的继承类
C. frame 的参考系是父视图坐标，bounds 的参考系是自身的坐标
D. frame 的参考系是自身坐标，bounds 的参考系是父视图的坐标

47. UIViewController在显示过程中，各个方法的调用顺序是：[单选题]

A. init -> viewDidLoad -> viewDidAppear -> viewDidUnload
B. init -> viewDidAppear -> viewDidLoad -> viewDidUnload
C. init -> viewDidLoad -> viewDidUnload -> viewDidAppear
D. init -> viewDidAppear -> viewDidUnload -> viewDidLoad

48. 使用imageNamed方法创建UIImage对象时，与普通的init方法有什么区别？[单选题]

A. 没有区别，只是为了方便
B. imageNamed方法只是创建了一个指针，没有分配其他内存
C. imageNamed方法将图片加载到内存中后不再释放
D. imageNamed方法将使用完图片后立即释放

49. 在没有navigationController的情况下，要从一个ViewController切换到另一个ViewController应该：[单选题]

A. [self.navigationController pushViewController:nextViewController animated:YES];
B. [self.view addSubview:nextViewController.view];
C. [self pushViewController:nextViewController animated:YES];
D. [self presentModalViewController:nextViewController animated:YES];

50. 什么是key window？[单选题]

A. App中唯一的那个UIWindow对象
B. 可以指定一个key的UIWindow
C. 可接收到键盘输入等事件的UIWindow
D. 不可以隐藏的那个UIWindow对象

51. UIButton从子类到父类一次继承自：[单选题]

A. UIView-> UIViewController->UIController
B. UIResponder-> UIControl-> UIView
C. UIControl-> UIResponder->UIViewController
D. UIControl-> UIView-> UIResponder

52. 在iOS坐标系统中，坐标的原点位于:[单选题]

A. 左下角
B. 右上角
C. 左上角
D. 右下角

53. Storyboard故事板功能是在哪个iOS版本发布的？[单选题]

A. iOS 4
B. iOS 5
C. iOS 6
D. iOS 7

54. 以下哪种标记表示来自Storyboard的连接方法？[单选题]

A. @IBMethod
B. @IBOutlet
C. @IBAction
D. @IBProperty

55. CALayer 是被定义在哪个框架中的？[单选题]

A. UIKit
B. Foundation
C. QuartzCore
D. Core Image

56. UIColor、UIImage 是定义在哪个框架中的？[单选题]

A. Foundation
B. CoreGraphics
C. UIKit
D. Core Image

57. UIKit框架位于iOS系统架构的哪一层？[单选题]

A. 核心操作系统层（Core OS layer）
B. 核心服务层（Core Services layer）
C. 媒体层（Media layer）
D. 可触摸层（Cocoa Touch layer）

58. iPhone X的分辨率比例是：[单选题]

A. 4:3
B. 16:9
C. 18:9
D. 5:3

59. iOS11之前导航栏的默认高度为：[单选题]

A. 32Pt
B. 48Pt
C. 64Pt
D. 96Pt

60. iOS11之后如果设置了prefersLargeTitles = YES，则导航栏的高度为：[单选题]

A. 32Pt
B. 48Pt
C. 64Pt
D. 96Pt

61. 以下哪个属性是iOS11开始引入的，用来控制safeAreaInsets是否加到layoutMargins上？[单选题]

A. layoutMargins
B. directionalLayoutMargins
C. preservesSuperviewLayoutMargins
D. insetsLayoutMarginsFromSafeArea

62. 以下哪个属性是iOS8开始引入的，用于指定视图和它的子视图之间的边距？[单选题]

A. layoutMargins
B. directionalLayoutMargins
C. preservesSuperviewLayoutMargins
D. insetsLayoutMarginsFromSafeArea

63. 下面哪些方法是UITableView的delegate方法：[多选题]

A. - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section;
B. - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath;
C. - (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath;
D. - (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath;

64. 以下关于UIView和CALayer描述正确的是：[多选题]

A. UIView之所以能够显示，是因为其内部的CALayer层对象。
B. UIView真正的绘图部分，由一个CALayer类来管理。
C. 无需通过CALayer，即可给UIView添加阴影、圆角大小、边框宽度和颜色等。
D. 访问UIView跟绘图和坐标有关的属性，例如frame、bounds等，实际上内部都是在访问它所包含的CALayer的相关属性。

65. 当将UIView的alpha属性设置为0之后，以下描述正确的是：[多选题]

A. 当前的UIView及其子视图都会被隐藏，而不管子视图的alpha值为多少。
B. 当前的UIView会被隐藏，而子视图不会受到父视图alpha值的影响。
C. 当前的UIView不会从响应者链中移除，仍然可以授受触摸事件。
D. 当前的UIView会从响应者链中移除，而响应者链中的下一个会成为第一响应者。

66. 当将UIView的isHidden属性设置为true之后，以下描述正确的是：[多选题]

A. 当前的UIView及其子视图都会被隐藏，而不管子视图的hidden值为多少。
B. 当前的UIView会被隐藏，而子视图不会受到父视图isHidden值的影响。
C. 当前UIView会从响应者链中移除，而响应者链中的下一个会成为第一响应者。
D. 当前的UIView不会从响应者链中移除，仍然可以授受触摸事件。

67. 关于UIView的frame和bounds属性，以下描述正确的是：[多选题]

A. UIView的frame(origin, size)属性：定义了一个矩形，描述一个UIView的大小和在父坐标系的位置。
B. UIView的frame(origin, size)属性：定义了一个矩形，描述一个UIView的大小和在自身坐标系的位置。
C. UIView的bounds(origin, size)属性：定义了一个矩形，描述一个UIView的大小和自身坐标系原点的位置。
D. UIView的bounds(origin, size)属性：定义了一个矩形，描述一个UIView的大小和父坐标系原点的位置。

68. 通过CGAffineTransform仿射转换，可以对UIView视图进行哪些变换操作？[多选题]

A. 缩放
B. 旋转
C. 隐藏
D. 位移

69. UIViewController包含以下哪些功能？[多选题]

A. UIViewController类负责创建其管理的视图，并在内存较低时将它们从内容中移出。
B. 可以通过视图控制器来将新的视图内容，以模态窗口的方式显示在当前视图的上方。
C. 视图控制器可以响应设备的方向变化，对视图进行相应的调整，以适应新的方向。
D. 一些特殊的视图控制器，如导航视图控制器、标签视图控制器，可使视图的管理更加方便和规范。

70. 关于UITabBarController，以下描述正确的是：[多选题]

A. UITabBarController不能添加到别的视图控制器中。
B. UITabBarController可以添加到别的视图控制器中。
C. UITabBarController的Tab bar可以放置在屏幕的底部和顶部。
D. UITabBarController的Tab bar只能放置在屏幕的底部。

71. 关于UINavigationController，以下描述正确的是：[多选题]

A. UINavigationController导航控制器用于显示多屏并具有一定层次结构的内容，是构建分层应用程序的主要工具。
B. UINavigationController维护了一个视图控制器栈，所有的子视图控制器都处于堆栈中。
C. 和UITabBarController相比，UINavigationController是作为栈来实现的，它更适合用于处理和显示分层的数据。
D. UINavigationController导航控制器在管理、切入和切出多个内容页方面，与UITabBarController类似。

72. UIControl类的UIControlEvent主要包含哪三种事件？[多选题]

A. 基于通知的控件事件
B. 基于值的控件事件
C. 基于触摸的控件事件
D. 基于编辑的控件事件

73. 以下关于NSLineBreakMode的枚举成员，描述正确的是：[多选题]

A. ByWordWrapping表示以单词为显示单位显示，后面部分省略不显示
B. ByCharWrapping表示以字符为显示单位显示，后面部分省略不显示
C. ByClipping表示剪切与文本宽度相同的内容长度，后半部分被删除
D. ByTruncatingHead表示开头省略，显示尾部文字内容

74. UITableView可用于以下哪种场合？[多选题]

A. 方便用户浏览具有层次结构的数据
B. 提供一个可供选择的列表
C. 显示一系列数据的索引列表
D. 利用分组结构，展示细节信息和更详细的设置

75. 开发者可以对UItableViewCell进行哪些操作？[多选题]

A. 自定义单元格的文字标题、图标、背景等属性。
B. 对单元格进行选择和高亮操作。
C. 配置单元格的附加视图。
D. 对单元格进行添加、删除、改变位置等操作。

76. 以下关于UICollectionView和UICollectionViewController，描述正确的是：[多选题]

A. UICollectionView 和 UICollectionViewController 类是iOS 6新引进的API。
B. 标准的UICollectionView对象包括三个部分：Cells、Supplementary Views、Decoration Views，这三个部分都是UIView的子类。
C. UICollectionView类负责管理数据的有序集合，并使用自定义布局的模式来呈现这些数据。
D. UICollectionView除了多列布局之外，还可以实现平铺的布局、圆形的布局和更多的自定义布局。

77. UIGestureRecognizer类包含以下哪些手势？[多选题]

A. UIPanGestureRecognizer
B. UIRotationGestureRecognizer
C. UILongPressGestureRecognizer
D. UISwipeGestureRecognizer

78. 以下对Storyboard故事板功能描述正确的是：[多选题]

A. 通过简单地拖曳，往视图控制器添加各种界面控件，可以降低学习的难度，提高开发工作的效率和趣味，使开发工作更加人性化。
B. 可以更好地查看和理解项目中所有页面的外观，以及页面之间的连接跳转关系。
C. Storyboard快速实现了视图之间的跳转关系。
D. 通过重写prepareForSegue方法，将视图控制器的所有跳转逻辑都汇聚一处，方便进行统一的界面跳转和参数传递。

79. 以下对Storyboard故事板功能描述正确的是：[多选题]

A. Storyboard通过Cell ProtoType原型，以及Static静态Cell的特性，让表格控制器实现起来更容易。
B. 通过Storyboard可以通过拖曳建立约束关系，从而快速实现各分辨率的适配。
C. 当Storyboard包含越来越多的视图控制器时，打开Storyboard将越来越慢。
D. 通过将常见的功能比如注册、分享模块独立成一个Storyboard，可以实现功能的模块化和重复利用。

80. 使用Single View Application模板创建的项目，将自带哪两个Storyboard文件？[多选题]

A. Home.storyboard
B. Main.storyboard
C. Index.storyboard
D. LaunchScreen.storyboard

81. 通过操作 CALayer 对象，可以很方便地调整 UIView 的哪些界面属性？[多选题]

A. 阴影
B. 圆角
C. 边框
D. 渐变色

82. iOS的用户界面有哪几种创建方式？[多选题]

A. Xib
B. Storyboard
C. Photoshop
D. 代码创建

83. 以下关于iPhone X尺寸描述正确的是：[多选题]

A. 高度增加了145pt，变成812pt
B. 状态栏高度由20pt变成44pt
C. 物理分辨率为1125px * 2436px
D. iPhoneX屏幕是5.8英寸

1 : A	2 : B	3 : B	4 : A	5 : A	6 : A	7 : A	8 : B
9 : A	10 : A	11 : A	12 : B	13 : A	14 : A	15 : A	16 : A
17 : A	18 : B	19 : A	20 : A	21 : B	22 : A	23 : B	24 : A
25 : A	26 : B	27 : A	28 : A	29 : B	30 : B	31 : A	32 : A
33 : A	34 : A	35 : A	36 : A	37 : C	38 : C	39 : C	40 : C
41 : A	42 : B	43 : A	44 : B	45 : B	46 : C	47 : A	48 : C
49 : D	50 : D	51 : D	52 : C	53 : B	54 : C	55 : C	56 : C
57 : D	58 : C	59 : C	60 : D	61 : D	62 : A	63 : C,D	64 : A,B,D
65 : A,D	66 : A,C	67 : A,C	68 : A,B,D	69 : A,B,C,D	70 : A,C	71 : A,B,C,D	72 : B,C,D
73 : A,B,C,D	74 : A,B,C,D	75 : A,B,C,D	76 : A,B,C,D	77 : A,B,C,D	78 : A,B,C,D	79 : A,B,C,D	80 : B,D
81 : A,B,C,D	82 : A,B,D	83 : A,B,C,D					

四.iOS篇

共有52题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 为App设置关键词，如果关键词包含竞品的名称，则关键词会被屏蔽？[判断题]

A. 正确

B. 错误

2. 为App设置关键词，关键词不需要包含app的名称？[判断题]

A. 正确

B. 错误

3. 如果App审核被拒的原因是Meta信息造成的，则不需要重新提交IPA文件吗？[判断题]

A. 正确

B. 错误

4. 在iOS App中实体物品的购买可以使用支付宝？[判断题]

A. 正确

B. 错误

5. 下载安装量无论是在App Store还是在Google Play，都是导致App排名上升或者下跌的主要因素？[判断题]

A. 正确

B. 错误

6. 在100字符长度的关键词列表中，越靠前的关键词权重越大？[判断题]

A. 正确

B. 错误

7. 在App的标题、子标题、描述文字等出现安卓或Andiord字样，有可能在审核时导致App被拒吗？[判断题]

A. 不可能

B. 很有可能

8. 个人开发者账号可以在App Store发布金融类应用吗？[判断题]

A. 可以

B. 不可以

9. 应用使用了私有API，会在审核时被拒吗？[判断题]

A. 会

B. 不会

10. 应用名称、安装包等地方包含test、demo等字样，会在审核时被拒吗？[判断题]

A. 会

B. 不会

11. 如果应用程序审核被拒并显示Binary Rejected，此时需要重新上传IPA文件吗？[判断题]

A. 需要

B. 不需要

12. 如果应用程序审核被拒并显示Metadata Rejected，此时需要重新上传IPA文件吗？[判断题]

A. 需要

B. 不需要

13. 苹果对连续性订阅抽成是15%？[判断题]

A. 正确

B. 错误

14. 开发者可以直接回复用户在App Store中的评论吗？[判断题]

A. 可以

B. 不可以

15. APP名称、截图和预览中包含价格信息（免费、打折）将无法上架 App Store？[判断题]

A. 正确

B. 错误

16. 在iOS11上，如果APP在启动时图标的四周出现黑色，是因为图标的四角的圆角，并且周围为透明像素。[判断题]

A. 正确

B. 错误

17. 在Apple App Store中，应用程序刚发布的前24小时的下载量是最大的？[判断题]

A. 正确
B. 错误

18. 您的应用如果处于等待开发者发布（Pending Developer Release）状态，将无法测试内购项目？[判断题]

A. 正确
B. 错误

19. iOS后台运行是在哪个系统版本才开始支持：[单选题]

A. iOS 3.0
B. iOS 4.0
C. iOS 5.0
D. iOS 6.0

20. 下列UIView的方法中，哪一个在iOS 5.0前后的系统调用机制不同：[单选题]

A. addSubView
B. layoutSubView
C. drawRect
D. removeFromSuperView

21. Siri实现了语音控制输入的功能。用户可以通过Siri技术，使用语音提问的方式进行人机交互。Siri的引入是从哪个iOS版本开始的？[单选题]

A. iOS 4
B. iOS 5
C. iOS 6
D. iOS 7

22. 从哪个iOS版本开始，系统UI从拟物风格转换为扁平化风格？[单选题]

A. iOS 5
B. iOS 6
C. iOS 7
D. iOS 8

23. Apple Pay是在哪个iOS版本中开始和大家见面的？[单选题]

A. iOS 6
B. iOS 7
C. iOS 8
D. iOS 9

24. 从哪个iOS版本开始，苹果开放了对第三方输入法的支持？[单选题]

A. iOS 7
B. iOS 8
C. iOS 9
D. iOS 10

25. Split View和画中画功能最早是在哪个iOS版本中引入的？[单选题]

A. iOS 7
B. iOS 8
C. iOS 9
D. iOS 10

26. 针对中国用户，苹果在哪个iOS版本中，开始对电话功能进入了十分体贴的优化，增加骚扰电话识别功能。[单选题]

A. iOS 7
B. iOS 8
C. iOS 9
D. iOS 10

27. ARKit增强现实功能和CoreML机器学习功能在哪个iOS版本中引入的？[单选题]

A. iOS 11
B. iOS 10
C. iOS 9
D. iOS 8

28. ARC自动引用计数和iCloud是在哪个iOS版本中新增的？[单选题]

A. iOS 7
B. iOS 6
C. iOS 5
D. iOS 4

29. 哪个iOS版本增加了对Bit 64的支持和引入了TextKit框架？[单选题]

A. iOS 7
B. iOS 6
C. iOS 5
D. iOS 4

30. 哪个iOS版本带来了Size class和Autolayout自动布局功能？[单选题]

A. iOS 7
B. iOS 8
C. iOS 9
D. iOS 10

31. 3D Touch和iPad分屏技术是在哪个iOS版本中开始引入的？[单选题]

A. iOS 7
B. iOS 8
C. iOS 9
D. iOS 10

32. 苹果在哪个iOS版本中向开发者开放了SiriKit框架？[单选题]

A. iOS 8
B. iOS 9
C. iOS 10
D. iOS 11

33. 作为推广ApplePay的一种策略，苹果在哪个iOS版本中，向开发者开放了NFC(Near field communication)功能？[单选题]

A. Xcode 8
B. Xcode 9
C. Xcode 10
D. Xcode 11

34. Core Image图像处理框架是从哪个iOS 版本起加入进来的？[单选题]

A. iOS 5
B. iOS 6
C. iOS 7
D. iOS 8

35. 自哪个版本的iOS开始，Apple为用户带来了炫酷的毛玻璃效果？[单选题]

A. iOS 6.0
B. iOS 7.0
C. iOS 8.0
D. iOS 9.0

36. Storyboard故事板功能是在哪个iOS版本发布的？[单选题]

A. iOS 4
B. iOS 5
C. iOS 6
D. iOS 7

37. iOS11之前导航栏的默认高度为：[单选题]

A. 32Pt
B. 48Pt
C. 64Pt
D. 96Pt

38. iOS11之后如果设置了prefersLargeTitles = YES，则导航栏的高度为：[单选题]

A. 32Pt
B. 48Pt
C. 64Pt
D. 96Pt

39. App Store允许开发者上传几张应用程序的截图?[单选题]

A. 3
B. 4
C. 5
D. 6

40. 以下哪个属性是iOS11开始引入的，用来控制safeAreaInsets是否加到layoutMargins上？[单选题]

A. layoutMargins
B. directionalLayoutMargins
C. preservesSuperviewLayoutMargins
D. insetsLayoutMarginsFromSafeArea

41. 以下哪个属性是iOS8开始引入的，用于指定视图和它的子视图之间的边距？[单选题]

A. layoutMargins
B. directionalLayoutMargins
C. preservesSuperviewLayoutMargins
D. insetsLayoutMarginsFromSafeArea

42. 关于iOS程序后台运行，下面说法正确的有：[多选题]

A. 程序可以在后台播放音乐
B. 程序可以在后台收集用户位置信息
C. 程序可以在后台运行VOIP服务
D. 程序可以在后台发送HTTP通讯

43. 关于iOS，以下说法正确的是？[多选题]

A. iOS是Apple公司推出的一款操作系统，是用于Apple移动设备的移动操作系统。
B. 由于最初是设计给iPhone使用的，所以该系统原名为iPhone OS，即“iPhone运行OS X”。
C. iOS系统使用了和macOS一样的Unix内核。
D. iOS系统可以应用在iWatch和iPod上。

44. 关于iOS开发，以下说法正确的是？[多选题]

A. 采用iOS系统的iPhone屏幕较小，只是把需要显示给用户的内容合理地组织在一块小小的屏幕上，所以需要设计者进行精心的设计和排版。
B. iOS采用手指触摸的方式进行人机交互，所以要尽可能使按钮等交互控件的尺寸保持在44点以上，以避免误操作。
C. 运行iOS系统的移动设备，通常内存在512MB～2GB之间。用户需要在应用中合理地使用多媒体素材，保证应用不会因太耗内存而被系统自动关掉。
D. 作为运行在移动设备上的应用，需要尽可能降低电量的消耗。比如及时关闭地理定位服务，减少不必要的网络请求，尽量避免以轮询的方式工作。

45. 关于iOS开发，以下说法正确的是？[多选题]

A. 一个App作为一个程序束bundle存在，App只可以访问其资源束之内的文件夹或其他资源文件。
B. 在iOS中运行的应用，可以访问移动设备自带的加速计、陀螺仪、地理定位设备、蓝牙、相机等。
C. iOS应用很少使用菜单进行页面之间的跳转，而是通常采用导航控制器或标签控制器进行页面之间的导航。
D. iOS系统中的应用，没有最小化和关闭按钮。用户通过按下设备底部的Home键，退出正在运行的应用，应用退出后仍然会在内存中保留一段时间。

46. 苹果的iOS系统采用了以下哪些严格的安全机制？[多选题]

A. 代码签名
B. 权限隔离
C. 可信启动链
D. 沙盒执行环境

47. 以下哪种情况会导致审核失败？[多选题]

A. 应用出现崩溃、加载失败等非常明显的Bug
B. 应用描述、截图等与应用功能严重不符
C. 错误使用抽奖、竞猜等促销方式
D. 包含虚假、误导用户的信息或功能

48. 应用程序在审核时被拒，可以分为哪两种情况？[多选题]

A. Binary Rejected
B. App Rejected
C. Metadata Rejected
D. Game Rejected

49. 除了从 AppStore 下载，我们还可以从哪些渠道安装一个 App？[多选题]

A. 开发 App 时可以直接把开发中的应用安装进手机进行调试。

B. In-House 企业内部分发，可以直接安装企业证书签名后的 APP。

C. AD-Hoc 相当于企业分发的限制版。

D. 使用开发者证书打包，并将包安装在开发者证书指定的设备上。

50. 获得苹果推荐的App需要包含哪些要素？[多选题]

A. 质量为上：获得苹果推荐的首要的条件便是产品质量。

B. 关注度：设计新颖，明确自己能传达给用户什么内容，同时具有独特的吸引力

C. 商业模式：适当的商业模式和价格，最好是和同类游戏相比有着独具一格的商业模式

D. 通用性：对于各种规格设备的支持，各个地区的本地化

51. 以下关于TestFlight描述正确的是：[多选题]

A. 使用 TestFlight，你可以向测试人员发布你 App 的 prerelease 版本来收集反馈信息。

B. TestFlight是一个必选功能，只有在使用它之后，才可以提交app到 appStore。

C. TestFlight是免费的。

D. TestFlight 只支持了 iOS 平台的 App，每个开发者账号最多只能测试10款不同的 App。

52. 在iOS 8里，苹果发布了一个新的接口叫做Metal，以下对Metal描述正确的是：[多选题]

A. Metal和OpenGL ES相似，它也是一个底层API。

B. Metal负责和3D绘图硬件交互。

C. Metal不是跨平台的。

D. Metal是一个支持GPU加速3D绘图的API。

1 : A	2 : A	3 : A	4 : A	5 : A	6 : A	7 : B	8 : B
9 : A	10 : A	11 : A	12 : B	13 : A	14 : A	15 : A	16 : A
17 : A	18 : A	19 : A	20 : B	21 : B	22 : C	23 : C	24 : B
25 : C	26 : D	27 : A	28 : C	29 : A	30 : B	31 : C	32 : C
33 : D	34 : A	35 : B	36 : B	37 : C	38 : D	39 : C	40 : D
41 : A	42 : A,B,C	43 : A,B,C	44 : A,B,C,D	45 : A,B,C,D	46 : A,B,C,D	47 : A,B,C,D	48 : A,C
49 : A,B,C,D	50 : A,B,C,D	51 : A,C,D	52 : A,B,C,D				

五.操作篇

共有68题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 使用iPhone模拟器可以测试相机、传感器等功能？[判断题]

A. 正确

B. 错误

2. 模拟器可以支持长按操作？[判断题]

A. 正确

B. 错误

3. 模拟器支持3D Touch操作？[判断题]

A. 正确

B. 错误

4. 程序单元是应用程序的最小可测试部件，最小可测试部件通常是包括基类、抽象类或者子类中的方法？[判断题]

A. 正确

B. 错误

5. 单元测试的关键组件是测试用例，测试用例可以在最低可测试的单元对代码进行测试？[判断题]

A. 正确

B. 错误

6. XCTest框架可以进行单元测试，但是无法进行性能测试？[判断题]

A. 正确

B. 错误

7. Xcode的UI测试功能能够自动记录你在应用程序界面中的每一个操作步骤，它可以将每一步的操作自动生成对应的代码吗？[判断题]

A. 可以

B. 不可以

8. Git和SVN等其它版本控制系统的一个重要不同之处，就是Git有暂存区?[判断题]

A. 正确

B. 错误

9. Git中几乎所有的操作都是可逆的？[判断题]

A. 正确
B. 错误

10. 可以为iPhone和iPad指定不同的LaunchScreen.storyboard故事板？[判断题]

A. 正确
B. 错误

11. LaunchScreen.storyboard故事板可以使用UIView或者UIViewController，而自定义的子类则不可以使用。[判断题]

A. 正确
B. 错误

12. 在 Xcode 9 中可以同时运行多个模拟器吗？[判断题]

A. 可以
B. 不可以

13. 在 Xcode 9 中可以任意调节模拟器的显示比例吗？[判断题]

A. 可以
B. 不可以

14. 使用Xcode创建工程时，支持同时创建的版本管理库是：[单选题]

A. Subversion
B. Mercurial
C. Git
D. Concurrent Versions System

15. 需要在手动管理内存分配和释放的Xcode项目中引入和编译用ARC风格编写的文件，需要在文件的Compiler Flags上添加参数：[单选题]

A. -shared
B. -fno-objc-arc
C. -fobjc-arc
D. -dynamic

16. 使用Xcode9的模拟器，如果需要清除模拟器中的所有应用程序，可以如何操作？[单选题]

A. 使用[Hardward > Restart]命令
B. 使用[Hardward > Erase All Content and Settings...]命令
C. 使用[File > Close Window]命令
D. 使用[Hardward > Lock]命令

17. 按下键盘上的哪个键，可以在模拟器中模拟双指触摸？[单选题]

A. Command
B. Option
C. Shift
D. Control

18. Instruments中的哪个工具，可以显示内存中的对象占用曲线，内存当前分配给了哪些对象，被分配内存的对象数量，以及由哪些函数进行对象的创建？[单选题]

A. Leaks
B. System Trace
C. Zombies
D. Allocations

19. Instruments中的哪个工具，可以查看内存泄露的对象数量，每个泄露对象的类型、内存地址和占用大小等信息?[单选题]

A. Leaks
B. System Trace
C. Zombies
D. Allocations

20. 使用CocoaPods安装第三方开源框架之后，双击以下哪个文件，来打开编辑安装后的项目。[单选题]

A. projectName.xcodeproject
B. projectName.xcworkspace
C. projectName.storyboard
D. projectName.plist

21. 当增加或删除了某个第三方开源框架，或第一次增加Podfile到工程时，你需要使用以下哪个命令来安装框架？[单选题]

A. pod install
B. pod update
C. pod update --no-repo-update
D. pod install --no-repo-update

22. 写入SQLite数据库，从数据库中取出 char 类型，当 char 类型有表示中文字符时，会出现乱码，这是因为数据库的默认编码方式为：[单选题]

A. ASCII
B. UTF-8
C. GB2312
D. GBK

23. 以下哪条sqlite语句，可以执行非查询的 sql 语句？[单选题]

A. sqlite3_open()
B. sqlite3_exec()
C. Sqlite3_step()
D. Sqlite3_close()

24. 以下哪个快捷键，可以左移一行或多行选中的代码？[单选题]

A. Command + [
B. Command +]
C. Command + <
D. Command + >

25. 以下哪个快捷键，可以隐藏或显示左侧的项目导航区？[单选题]

A. Control + 1
B. Control + 0
C. Command + 1
D. Command + 0

26. 以下哪个快捷键，可以隐藏或显示右侧的工具区(Utilities)？[单选题]

A. Command + Alt + 0
B. Command + Alt + 1
C. Command + 0
D. Command + 1

27. 以下哪个快捷键，可以上移一行或多行选择的代码？[单选题]

A. Option + Command + <
B. Option + Command + >
C. Option + Command + [
D. Option + Command +]

28. 以下哪个快捷键，可以新建一个项目？[单选题]

A. Command + Shift + A
B. Command + Shift + N
C. Command + N
D. Command + A

29. 以下哪个快捷键，可以新建一个文件？[单选题]

A. Command + Shift + A
B. Command + Shift + N
C. Command + N
D. Command + A

30. 以下哪个快捷键，可以在项目中搜索一个文件？[单选题]

A. Command + Shift + F
B. Command + Shift + S
C. Command + F
D. Command + S

31. 以下哪个快捷键，可以对当前项目执行Clean操作？[单选题]

A. Command + Shift + F
B. Command + Shift + K
C. Command + Shift + H
D. Command + Shift + N

32. 以下哪个快捷键，可以快速编译并运行项目？[单选题]

A. Command + B
B. Command + S
C. Command + N
D. Command + R

33. 以下哪个快捷键，可以快速对代码进行注释和取消注释？[单选题]

A. Command + /
B. Command + \
C. Command + C
D. Command + T

34. 以下哪个快捷键，可以快速在当前行插入或清除断点？[单选题]

A. Command + /
B. Command + \
C. Command + C
D. Command + T

35. 以下哪个快捷键，可以将光标快速移到文件头部？[单选题]

A. Command + 上箭头
B. Command + 下箭头
C. Command + 左箭头
D. Command + 右箭头

36. 以下哪个快捷键，可以将光标快速移到文件尾部？[单选题]

A. Command + 上箭头
B. Command + 下箭头
C. Command + 左箭头
D. Command + 右箭头

37. 以下哪个快捷键，可以将光标快速移到行首？[单选题]

A. Command + 上箭头
B. Command + 下箭头
C. Command + 左箭头
D. Command + 右箭头

38. 以下哪个快捷键，可以将光标快速移到行尾？[单选题]

A. Command + 上箭头
B. Command + 下箭头
C. Command + 左箭头
D. Command + 右箭头

39. 以下哪个快捷键，可以终止编译或模拟器的运行？[单选题]

A. Command + ,
B. Command + .
C. Command + '
D. Command + ~

40. 以下哪个快捷键，可以在同一行上将光标向右移动？[单选题]

A. Control + M
B. Control + B
C. Control + F
D. Control + P

41. 以下哪个快捷键，可以在同一行上将光标向左移动？[单选题]

A. Control + M
B. Control + B
C. Control + F
D. Control + P

42. 以下哪个快捷键，可以将光标移动到前一行？[单选题]

A. Control + M
B. Control + B
C. Control + F
D. Control + P

43. 以下哪个快捷键，可以将光标移动到后一行？[单选题]

A. Control + N
B. Control + B
C. Control + F
D. Control + P

44. 以下哪个快捷键，可以将光标移动到本行的行首？[单选题]

A. Control + N
B. Control + A
C. Control + F
D. Control + P

45. 以下哪个快捷键，可以将光标移动到本行的行尾？[单选题]

A. Control + N
B. Control + A
C. Control + N
D. Control + P

46. 以下哪个快捷键，可以删除光标右边的字符？[单选题]

A. Control + N
B. Control + A
C. Control + N
D. Control + D

47. 以下哪个快捷键，可以删除光标所在行光标后面的代码，便于您重写行尾代码。[单选题]

A. Control + N
B. Control + K
C. Control + N
D. Control + D

48. 以下哪个快捷键，可以将显示区域移到光标显示的中心位置？[单选题]

A. Control + N
B. Control + A
C. Control + L
D. Control + D

49. 以下哪个快捷键，可以打开文件跳转列表，方便查看和打开历史文件？[单选题]

A. Control + 1
B. Control + 2
C. Control + 3
D. Control + 4

50. 以下哪个快捷键，可以打开与当前文件处于同一文件夹下的所有文件和文件夹列表？[单选题]

A. Control + 4
B. Control + 5
C. Control + 6
D. Control + 7

51. 以下哪个快捷键，可以显示当前代码文件的所有方法列表？[单选题]

A. Control + 4
B. Control + 5
C. Control + 6
D. Control + 7

52. 以下哪个快捷键，可以选择光标右侧的代码？[单选题]

A. Control + Shift + E
B. Control + Shift + A
C. Control + Shift + F
D. Control + Shift + H

53. 以下哪个快捷键，可以选择光标左侧的代码？[单选题]

A. Control + Shift + E
B. Control + Shift + A
C. Control + Shift + F
D. Control + Shift + H

54. 以下哪个快捷键，可以复制一个标签窗口？[单选题]

A. Command＋A
B. Command＋E
C. Command＋T
D. Command＋J

55. 以下哪个快捷键，可以打开辅助编辑器？[单选题]

A. Command + Shift + Enter
B. Command + Control + Enter
C. Command + Enter
D. Command + Alt + Enter

56. 以下哪个快捷键，可以关闭打开的辅助编辑器？[单选题]

A. Command + Shift + Enter
B. Command + Control + Enter
C. Command + Enter
D. Command + Alt + Enter

57. 以下哪个快捷键，可以通过键入关键词的方式，快速切换至某个文件？[单选题]

A. Command + Shift + O
B. Command + Shift + A
C. Command + Shift + H
D. Command + Shift + K

58. CALayer, PDFDocument, NSImageRep, CIRenderTask, CIContext & CIImage 这些类型的数据可以在 哪个版本的 Xcode 中直接看到数据内容，不再是简单的内存地址了？[单选题]

A. Xcode 6
B. Xcode 7
C. Xcode 8
D. Xcode 9

59. 在Instruments工具中，用于检查内存泄露的工具是：[多选题]

A. Counters
B. Allocations
C. Leaks
D. Energy Log

60. 可以通过代码进行写操作的文件或目录有？[多选题]

A. bundle目录
B. Documents目录
C. keychain
D. SQLite

61. 以下哪个操作可以获取模拟器的截图？[多选题]

A. 依次点击[File > Save Screen Shot]菜单命令。
B. 使用[Command + D]快捷键
C. 使用[Command + S]快捷键
D. 使用[Command + Shift + S]快捷键

62. 使用Xcode9的模拟器，如果需要退出当前的应用程序，可以如何操作？[多选题]

A. 使用[Command + Shift + H]快捷键
B. 使用[Hardware > Home]命令
C. 使用[Command + Shift + E]快捷键
D. 点击模拟器的Home键

63. 使用 Instruments提供的多种工具，可以对应用程序进行哪些操作？[多选题]

A. 追查代码中难以重现的问题。
B. 对应用程序进行性能分析。
C. 对应用程序进行压力测试。
D. 进行一般的系统级故障诊断。

64. 以下关于iOS开发的第三方依赖开源框架管理工具，说法正确的是：[多选题]

A. SMP现在仅仅支持Swift语言，不支持iOS和watch OS平台。
B. Carthage仅仅支持最低版本为iOS 8.0的工程。
C. CocoaPods则拥有最多的开源框架支持，并且可以支持Swift和Objective-C两种语言编写的Cocoa工程。
D. Carthage是一个用Swift编写的第三方依赖开源框架管理工具。

65. 如果你的操作不需要更新podspecs，执行哪些命令可以快速更新第三方开源框架？[多选题]

A. pod update
B. pod update --no-repo-update
C. pod install
D. pod install --no-repo-update

66. Git中的哪些命令需要在有网络的环境中执行：[多选题]

A. pull
B. commit
C. status
D. push

67. Git版本控制有哪三种状态？[多选题]

A. committed
B. updated
C. staged
D. modified

68. Xcode 9 支持哪些转换和重构？[多选题]

A. 添加缺少的协议要求
B. 为抽象方法添加缺少的重写
C. 提取到局部变量
D. 提取方法/表达式

1 : B	2 : A	3 : B	4 : A	5 : A	6 : B	7 : A	8 : A
9 : A	10 : B	11 : A	12 : A	13 : A	14 : C	15 : C	16 : B
17 : B	18 : D	19 : A	20 : B	21 : A	22 : A	23 : B	24 : A
25 : D	26 : A	27 : C	28 : B	29 : C	30 : A	31 : B	32 : D
33 : A	34 : B	35 : A	36 : B	37 : C	38 : D	39 : B	40 : C
41 : B	42 : D	43 : A	44 : B	45 : C	46 : D	47 : B	48 : C
49 : A	50 : B	51 : C	52 : A	53 : B	54 : C	55 : D	56 : C
57 : A	58 : D	59 : B,C	60 : B,C,D	61 : A,C	62 : A,B,D	63 : A,B,C,D	64 : A,B,C,D
65 : B,D	66 : A,D	67 : A,C,D	68 : A,B,C,D				

六.数据结构篇

共有23题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 局部变量是保存在栈区的，方法调用的实参也是保存在栈区的？[判断题]

A. 正确

B. 错误

2. 栈是向低地址扩展的数据结构，是一块连续的内存的区域？[判断题]

A. 正确

B. 错误

3. 堆是向高地址扩展的数据结构，是不连续的内存区域吗？[判断题]

A. 正确

B. 错误

4. 栈是由系统自动分配，速度较快，不会产生内存碎片？[判断题]

A. 正确

B. 错误

5. 堆是由alloc分配的内存，速度比较慢，而且容易产生内存碎片，不过用起来最方便。[判断题]

A. 正确

B. 错误

6. int a;是指一个整型数（An integer）？[判断题]

A. 正确

B. 错误

7. int *a；是指一个指向整型数的指针（ A pointer to an integer）？[判断题]

A. 正确

B. 错误

8. int **a;是一个指向指针的的指针，它指向的指针是指向一个整型数？（ A pointer to a pointer to an intege）[判断题]

A. 正确

B. 错误

9. int a[10];是指一个有10个整型数的数组？（An array of 10 integers）[判断题]

A. 正确

B. 错误

10. int *a[10];是指一个有10个指针的数组，该指针是指向一个整型数的。（An array of 10 pointers to integers）[判断题]

A. 正确

B. 错误

11. int (*a)[10];是指一个指向有10个整型数数组的指针（A pointer to an array of 10 integers）[判断题]

A. 正确

B. 错误

12. int (*a)(int);一个指向函数的指针，该函数有一个整型参数并返回一个整型数（A pointer to a function that takes an integer as an argument and returns an integer）[判断题]

A. 正确

B. 错误

13. int (*a[10])(int); 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（An array of ten pointers to functions that take an integer argument and return an integer）[判断题]

A. 正确

B. 错误

14. 堆和栈的区别正确的是：[单选题]

A. 对于栈来讲，我们需要手工控制，容易产生memory leak

B. 对于堆来说，释放工作由编译器自动管理，无需我们手工控制

C. 在Windows下,栈是向高地址扩展的数据结构，是连续的内存区域，栈顶的地址和栈的最大容量是系统预先规定好的。

D. 对于堆来讲，频繁的new/delete势必会造成内存空间的不连续，从而造成大量的碎片，使程序效率降低。

15. 下面四种内部排序算法中哪一种在最差情况下时间复杂度最高？[单选题]

A. 快速排序

B. 冒泡排序

C. 堆排序

D. 归并排序

16. 下面的数据结构中不属于线性结构的是：[单选题]

A. 栈

B. 链表

C. 二叉树

D. 线性表

17. 在一个二叉树上，第5层最多可以有的节点数是：[单选题]

A. 2
B. 8
C. 16
D. 32

18. 在长度为n的线性表上进行顺序查找，在最糟糕的情况下需要的比较次数是：[单选题]

A. n
B. 2n-1
C. 2n
D. n^2

19. 已知二叉树后序遍历序列是dabec，中序遍历序列是debac，它的前序遍历序列是：[单选题]

A. cedba
B. acbed
C. decab
D. deabc

20. 多线程中栈与堆是公有的还是私有的：[单选题]

A. 栈公有, 堆私有
B. 栈公有，堆公有
C. 栈私有, 堆公有
D. 栈私有，堆私有

21. 求以下程序段的输出int test(int x, int y){ x = x + y; return x * y;}int main(int argc, const char * argv[]){ @autoreleasepool { int x = 3, y = 10, z = test(x, y); NSLog(@"%d%d", x++, ++z); } return 0;}[单选题]

A. 331
B. 330
C. 431
D. 430

22. 求以下程序段的输出int func(int x){ int countx = 0; while (x) { countx++; x = x&(x-1); } return countx;}int main(int argc, const char * argv[]){ @autoreleasepool { NSLog(@"%d", func(2013)); } return 0;}[单选题]

A. 1
B. 5
C. 8
D. 9

23. main() { int a[5]={1,2,3,4,5}; int *ptr=(int *)(&a+1); printf("%d,%d",*(a+1),*(ptr-1)); }的答案是：[单选题]

A. 5,2
B. 2,5
C. 1,4
D. 4,1

1 : A	2 : A	3 : A	4 : A	5 : A	6 : A	7 : A	8 : A
9 : A	10 : A	11 : A	12 : A	13 : A	14 : D	15 : B	16 : C
17 : C	18 : A	19 : A	20 : C	21 : A	22 : D	23 : B	

七.多线程篇

共有60题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. Cocoa中的NSRunLoop类并不是线程安全的？[判断题]

A. 正确

B. 错误

2. Run loop的管理并不完全是自动的？[判断题]

A. 正确

B. 错误

3. Run loop同时也负责autorelease pool的创建和释放？[判断题]

A. 正确

B. 错误

4. 每个线程不仅仅在创建的过程中需要耗费时间，同时它也会占用一定内核的内存空间和应用程序的内存空间?[判断题]

A. 正确

B. 错误

5. 在iOS中，主线程的栈空间大小是可以修改的？[判断题]

A. 正确

B. 错误

6. iOS应用程序子线程的栈空间大小，会在线程的使用过程中逐渐增加？[判断题]

A. 正确

B. 错误

7. 子线程允许分配的栈空间大小，必须为4KB的整数倍？[判断题]

A. 正确

B. 错误

8. 线程的优先级属性threadPriority是一个0.0～1.0之间的浮点数，那么1.0表示最高的优先级？[判断题]

A. 正确

B. 错误

9. 线程的默认的优先级是0.5？[判断题]

A. 正确

B. 错误

10. 优先级较高的线程，一定比优先级较低的线程先执行？[判断题]

A. 正确

B. 错误

11. iOS应用程序的每条线程，都有唯一的RunLoop对象与之对应？[判断题]

A. 正确

B. 错误

12. 子线程的RunLoop需要手动创建？[判断题]

A. 正确

B. 错误

13. 主线程的RunLoop是自动创建并启动的？[判断题]

A. 正确

B. 错误

14. currentRunLoop是延迟加载的，只创建一次？[判断题]

A. 正确

B. 错误

15. 使用Thread创建的线程，将共享同一应用程序的部分内存空间，它们拥有对数据相同的访问权限？[判断题]

A. 正确

B. 错误

16. 为了协调多个线程对同一数据的访问，通常的做法是在访问之前加锁，加锁会导致一定的性能开销吗？[判断题]

A. 不会

B. 会

17. 当一个Lock对象成功调用lock方法后，在其调用unlock方法之前，任何线程都不能再对此Lock对象加锁？[判断题]

A. 正确

B. 错误

18. ConditionLock是一种带有条件的锁，可以根据条件对线程进行加锁？[判断题]

A. 正确

B. 错误

19. 也就是说一个线程可以对一个Recursive对象多次调用lock，只要解锁时调用相同次数的unlock方法便可。以上是RecursiveLock的特征吗？[判断题]

A. 不是

B. 是

20. 因为线程是消耗系统资源的，所以需要控制线程的并发数，以避免系统变慢？[判断题]

A. 是

B. 不是

21. 由于GCD基于工作单元而非像Thread那样基于运算，所以GCD可以控制诸如等待任务结束、监视文件描述符、周期执行代码以及工作挂起等任务?[判断题]

A. 正确

B. 错误

22. GCD的dispatch queue调度队列可以是并发的，也可以是串行的？[判断题]

A. 正确

B. 错误

23. dispatch_semaphore是GCD用来同步的一种方式吗？[判断题]

A. 是

B. 不是

24. NSRecursiveLock是一个递归锁，这个锁可以被同一线程多次请求，而不会引起死锁吗？[判断题]

A. 是

B. 不是

25. 请求加锁失败的话，OSSpinLock会一直轮询，等待时会消耗大量 CPU 资源，所以OSSpinLock不适用于较长时间的任务？[判断题]

A. 正确
B. 错误

26. NSLock 请求加锁失败的话，会先轮询，但一秒过后便会使线程进入 waiting 状态，等待唤醒。以上说法正确吗？[判断题]

A. 正确
B. 错误

27. pthread_mutex 是基于OC语言的多线程加互斥锁的方式吗？[判断题]

A. 是
B. 不是

28. 下面方法中，哪一个方法没有创建新的线程？[单选题]

A. + (void)detachNewThreadSelector:(SEL)selector toTarget:(id)target withObject:(id)argument;
B. - (id)initWithTarget:(id)target selector:(SEL)selector object:(id)argument;
C. - (void)performSelectorInBackground:(SEL)aSelector withObject:(id)arg;
D. - (void)performSelectorOnMainThread:(SEL)aSelector withObject:(id)arg waitUntilDone:(BOOL)wait;

29. 下列技术不属于多线程的是：[单选题]

A. Block
B. Thread
C. Operation
D. GCD

30. 线程和进程的区别不正确的是：[单选题]

A. 进程和线程都是由操作系统所体会的程序运行的基本单元
B. 线程之间有单独的地址空间
C. 进程和线程的主要差别在于它们是不同的操作系统资源管理方式
D. 线程有自己的堆栈和局部变量

31. 对于runloop的理解不正确的是：[单选题]

A. 每一个线程都有其对应的RunLoop
B. 默认非主线程的RunLoop是没有运行的
C. 在一个单独的线程中没有必要去启用RunLoop
D. 可以将NSTimer添加到runloop中

32. 以下多线程对int型变量x的操作，哪个不需要进行同步：[单选题]

A. x=y
B. x++
C. ++x
D. x=1

33. 多线程中栈与堆是公有的还是私有的：[单选题]

A. 栈公有, 堆私有
B. 栈公有，堆公有
C. 栈私有, 堆公有
D. 栈私有，堆私有

34. 下面关于线程管理错误的是：[单选题]

A. GCD所用的开销要比NSThread大
B. 可以在子线程中修改UI元素
C. NSOperationQueue是比NSthread更高层的封装
D. GCD可以根据不同优先级分配线程

35. Object C中的线程下面描述不正确的是：[单选题]

A. 使用NSThread创建、使用GCD的dispatch
B. 直接使用NSOperation,然后将其加入NSOperationQueue
C. 在主线程执行代码，方法是performSelectorOnMainThread,
D. 如果想延时执行代码可以用performSelector:onThread:withObject:waitUntilDone:

36. 对NSOperationQueue理解不正确的是：[单选题]

A. 存放NSOperation的集合类
B. 可以设置最大并发数
C. 放进去的线程会自动执行
D. 用户需要管理放进去的线程执行顺序

37. 在iOS中，主线程的栈空间大小为：[单选题]

A. 1MB
B. 4MB
C. 8MB
D. 32MB

38. 在iOS中，应用程序子线程的默认栈空间大小为：[单选题]

A. 256 KB
B. 512 KB
C. 1 MB
D. 4 MB

39. iOS子线程允许分配的最小栈空间是：[单选题]

A. 4 KB
B. 8 KB
C. 16 KB
D. 32 KB

40. 在线程自创建至消亡的整个生命周期中包含多个状态，各状态的顺序是：[单选题]

A. 创建状态 就绪状态 阻塞状态 运行状态 消亡状态
B. 创建状态 就绪状态 运行状态 阻塞状态 消亡状态
C. 就绪状态 创建状态 运行状态 阻塞状态 消亡状态
D. 就绪状态 创建状态 阻塞状态 运行状态 消亡状态

41. 子线程的RunLoop，当需要手动创建，调用以下哪个方法来启动？[单选题]

A. start()
B. fire()
C. new()
D. run()

42. 为了在同一个时间点内只执行一个线程，且当前一个线程结束后才执行第二个线程，需要将队列的maxConcurrentOperationCount设置为：[单选题]

B. 1
C. 2
D. 10

43. Grand Central Dispatch（GCD）是由Apple公司在哪个iOS版本时，推出的一个多核编程的解决方案。[单选题]

A. iOS 4.0
B. iOS 5.0
C. iOS 6.0
D. iOS 7.0

44. GCD的API很大程度上是和以下哪种技术配合使用的？[单选题]

A. delegate
B. block
C. kvo
D. kvc

45. Thread多线程技术是基于什么来实现多线程的？[单选题]

A. Thread线程
B. Queue队列
C. Block闭包
D. Task任务

46. Operation多线程技术是基于什么来实现多线程的？[单选题]

A. Thread线程
B. Queue队列
C. Block闭包
D. Task任务

47. Grand Central Dispatch多线程技术是基于什么来实现多线程的？[单选题]

A. Thread线程
B. Queue队列
C. Block闭包
D. Task任务

48. GCD Queue 分为哪三种？[多选题]

A. The main queue 主队列
B. Global queues 全局并发队列
C. 用函数 dispatch_queue_create创建的用户队列
D. Default queues 默认队列

49. 死锁有哪些必要条件？[多选题]

A. 互斥
B. 请求保持
C. 不可剥夺
D. 环路

50. 以下哪种方法可以处理死锁？[多选题]

A. 鸵鸟策略
B. 预防策略
C. 避免策略
D. 检测与解除死锁

51. 以下关于iOS开发中的多线程概念，描述正确的是：[多选题]

A. Multithreading多线程是指从软件或者硬件上，实现多个线程并发执行的技术。
B. 具有多线程能力的硬件，因为有硬件的支持，所以能够在同一时间内执行多于一个的线程，进而提升应用程序的整体处理性能。
C. 多线程能够同步完成多项任务，不是为了提高运行效率，而是为了提高资源使用效率，从而提高应用程序的效率。
D. 进程启动之后，一个最主要的线程被称为主线程，主线程会创建和管理应用程序中所有的UI元素。

52. Thread的创建分为显式和隐式两种类型，其中显式方式有哪两种：[多选题]

A. Thread.detachNewThreadSelector(selector: toTarget: with:)
B. object.performSelector(inBackground: with:)
C. object.performSelector(inBackground: afterDelay with:)
D. Thread.init(target: selector: object:)

53. iOS提供了哪些类型的线程锁：[多选题]

A. Lock
B. RecursiveLock
C. ConditionLock
D. UnLock

54. Cocoa operation 相关的类有哪些？[多选题]

A. Operation
B. NSLock
C. OperationQueue
D. NSThread

55. 以下关于并发、串行、同步和异步描述正确的是：[多选题]

A. 并发：多个任务同时执行
B. 串行：一个任务执行完成后，再执行下一个任务
C. 同步：在当前线程中执行任务，不会开启新线程
D. 异步：在新的线程中执行任务

56. GCD的dispatch queue调度队列有以下哪三种类型？[多选题]

A. The main queue 串行队列
B. Global queues 全局队列
C. 用户线程队列
D. 子线程队列

57. Global queues 全局队列是并发队列，它包含哪些优先级？[多选题]

A. 高
B. 中
C. 低
D. 后台

58. Timer定时器有哪两种功能？[多选题]

A. 在指定的时间执行指定的任务。
B. 在指定的时间范围内执行指定的任务。
C. 可以实现推送通知的任务。
D. 每隔一段时间执行指定的任务。

59. 以下保持线程同步的方式有哪些？[多选题]

A. @synchronized
B. NSLock
C. NSCondition
D. pthread_mutex

60. OSSpinLock 是一种自旋锁，它有哪三个方法？[多选题]

A. 加锁
B. 尝试加锁
C. 尝试解锁
D. 解锁

1 : A	2 : A	3 : A	4 : A	5 : B	6 : A	7 : A	8 : A
9 : A	10 : B	11 : A	12 : A	13 : A	14 : A	15 : A	16 : B
17 : A	18 : A	19 : B	20 : A	21 : A	22 : A	23 : A	24 : A
25 : A	26 : A	27 : B	28 : D	29 : A	30 : A	31 : C	32 : D
33 : C	34 : B	35 : B	36 : D	37 : A	38 : B	39 : C	40 : B
41 : D	42 : B	43 : A	44 : B	45 : A	46 : B	47 : D	48 : A,B,C
49 : A,B,C,D	50 : A,B,C,D	51 : A,B,C,D	52 : A,D	53 : A,B,C	54 : A,C	55 : A,B,C,D	56 : A,B,C
57 : A,B,C,D	58 : A,D	59 : A,B,C,D	60 : A,B,D				

八.网络篇

共有22题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 在LaunchScreen.storyboard故事板中，不能使用 UIWebView，因为这里是不能使用网络的。[判断题]

A. 正确
B. 错误

2. 通过 URL 我们可以定位一个远程 web 服务器上的资源位置，但是无法定位硬盘上的一个本地文件的路径？[判断题]

A. 正确
B. 错误

3. iOS开发中，HTTPS通讯是在什么位置来保障安全性：[单选题]

A. NSURLRequest方法里
B. NSURLRequest代理方法里
C. NSURLConnection方法里
D. NSURLConnection代理方法里

4. 关于NSURLConnection同步通讯和异步通讯，下面说法正确的是：[单选题]

A. 同步通讯是指发送数据后，不等接收方回应，接着发下一个数据
B. 异步通讯时会阻塞当前线程
C. 发送同步通讯时，系统会自动创建一个单独的线程
D. 一般不在主线程中使用同步通讯，因为会引起主线程阻塞

5. 关于iCloud，下面说法错误的是：[单选题]

A. 可以通过程序，将一个文档保存到用户的iCloud里
B. 用以通过程序，在iCloud里只保存一个键值对
C. 不能在模拟器上测试iCloud程序
D. 程序使用iCloud时，只需要使用相应的API即可，不需要额外的配置

6. 应用程序大小超过多少时，只能通过WiFi从App Store上下载：[单选题]

A. 5MB
B. 10MB
C. 20MB
D. 100MB

7. 即时聊天app不会采用的网络传输方式是：[单选题]

A. UDP
B. TCP
C. Http
D. FTP

8. 断点续传需要在请求头中添加的控制续传最重要的关键字是：[单选题]

A. range
B. length
C. type
D. size

9. post传输的最大文件限制为：[单选题]

A. 1G
B. 2G
C. 4G
D. 8G

10. 对于中国的客户来说，Map Kit使用的是来自哪个平台的数据？[单选题]

A. 百度
B. 高德
C. 谷歌
D. 苹果

11. MKMapCamera的heading属性值为多少，表示地图的顶部是正北方？[单选题]

B. 90
C. 180
D. 270

12. MKMapCamera的pitch属性值为多少，表示垂直往下看的角度，将产生标准的二维地图？[单选题]

B. 90
C. 180
D. 270

13. 以下哪些Cocoa Touch框架，用于网络访问：[多选题]

A. OpenAL
B. Bonjour
C. WebKit
D. BSD Sockets

14. AFNetworking网络框架是一个在iOS开发中使用非常多的网络开源库，适用于以下哪些平台？[多选题]

A. iOS
B. mac OS
C. watch OS
D. tv OS

15. 以下关于HTTP协议，描述正确的是：[多选题]

A. GET一般用于获取/查询资源信息
B. POST一般用于提交/更新资源信息
C. PUT一般用于增加资源信息
D. DELETE一般用于删除资源信息

16. MKMapView可以完成哪些任务？[多选题]

A. 显示指定地理位置的地图，如显示上海市的地图
B. 设置地图的显示方式
C. 在地图上对指定坐标位置做标记
D. 支持地图的放大和缩小

17. Map Kit地图显示方式有哪几种？[多选题]

A. 标准式的行政地图（会显示城市，街道等）
B. 标准的卫星地图
C. 卫星地图和混合地图（在卫星图上显示街道等名称）
D. 高清的卫星地图

18. 默认情况下，calloutView标注视图拥有哪几个部分？[多选题]

A. 标题
B. 副标题
C. 右辅助视图
D. 左辅助视图

19. 在使用MKMapCamera相机时，需要设置哪几个属性？[多选题]

A. 相机在地图上的坐标
B. 相机的观察方向
C. 相机和地图之间的角度
D. 以及相机在地图上方的高度

20. 如果需要获得一个网址中的参数，可以使用URL对象的哪些属性？[多选题]

A. parameterString
B. scheme
C. query
D. port

21. 现在的IM即时通讯协议包含哪些？[多选题]

A. SIP： IETF 的对话初始协议，是建立 VOIP 连接的 IETF 标准，而 VOIP 就是网络电话。
B. SIMPLE：即时通讯对话初始协议和表示扩展协议。
C. XMPP： 基于 XML 且开放的可扩展通讯和表示协议，常称为 Jabber 协议。
D. PRIM： 显示和即时通讯协议。

22. 以下哪些是WKWebView的优点？[多选题]

A. 拥有 60fps 滚动刷新率
B. 内置手势
C. 高效的 app 和 web 信息交换通道
D. 和 Safari 相同的 JavaScript 引擎

1 : A	2 : B	3 : D	4 : D	5 : D	6 : C	7 : D	8 : A
9 : C	10 : B	11 : A	12 : A	13 : B,C,D	14 : A,B,C,D	15 : A,B,C,D	16 : A,B,C,D
17 : A,B,C	18 : A,B,C,D	19 : A,B,C,D	20 : A,C	21 : A,B,C,D	22 : A,B,C,D		

九.多媒体篇

共有59题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 使用UIImageView的animationImages属性播放动画，图片序列中的图片需要具有相同的尺寸，如果序列中的图片具有不同的尺寸，可能会产生意料之外的动画效果？[判断题]

A. 正确

B. 错误

2. 使用System Sound Services播放音频，无法直接进行循环播放或立体声播放的控制？[判断题]

A. 正确

B. 错误

3. 使用AVAudioPlayer技术，可以控制播放的音量、设置立体声，还可以设置播放中声音的速率？[判断题]

A. 正确

B. 错误

4. AVAudioPlayer技术不支持进行声音播放的快进和后退？[判断题]

A. 正确

B. 错误

5. 在Quartz 2D中，使用含有Create或Copy的函数创建的对象，使用完后必须释放，否则将导致内存泄露？[判断题]

A. 正确

B. 错误

6. 和UIKit的坐标系一样，Quartz 2D 的坐标系原点也是在屏幕的左上角？[判断题]

A. 正确

B. 错误

7. 在 Quartz 2D 中，使用方法 CGContextAddLineToPoint 移动画笔到一个点来开始新的子路径？[判断题]

A. 正确

B. 错误

8. 每个CIFilter滤镜对象至少拥有一个输入参数，并且产生一个输出图像？[判断题]

A. 正确

B. 错误

9. CIContext对象可以基于GPU，也可以基于CPU？[判断题]

A. 正确

B. 错误

10. 当使用GPU进行渲染时，由于是使用Openg GL ES技术，所以可以获得比CPU更快的渲染速度，但是无法在后台进行操作？[判断题]

A. 正确

B. 错误

11. CIContext在默认情况下是使用GPU进行渲染的？[判断题]

A. 正确

B. 错误

12. CIContext在使用CPU渲染时，速度比GPU较慢。此时可以通过iOS的GCD技术在后台进行渲染，从而提升用户的体验吗。[判断题]

A. 可以

B. 不可以

13. CIDetector对象无法在视频中进行特征的搜索和检测？[判断题]

A. 正确

B. 错误

14. QuartzCore 框架和 CoreGraphics 框架是可以跨平台使用的，在 iOS 和 mac OS上都能使用？[判断题]

A. 正确

B. 错误

15. QuartzCore框架可以直接使用 UIImage、UIColor？[判断题]

A. 正确

B. 错误

16. CADisplayLink 是一个能让我们以和屏幕刷新率相同的频率将内容画到屏幕上的定时器？[判断题]

A. 正确

B. 错误

17. Core Animation动画框架可以用在iOS 平台，无法用在mac OS平台？[判断题]

A. 正确

B. 错误

18. Core Animation是可以直接作用在CALayer上的，所以它也可以作用在UIView上？[判断题]

A. 正确

B. 错误

19. Core Animation框架的动画执行过程都是在后台操作的，不会阻塞主线程？[判断题]

A. 正确

B. 错误

20. CAAnimation 是所有动画对象的父类，可以直接使用它，而无需使用它具体的子类？[判断题]

A. 正确

B. 错误

21. SceneKit可以与 Core Image，Core Animation，SpriteKit 等已有的图形框架相互整合及协作吗？[判断题]

A. 可以

B. 不可以

22. Scene Kit框架是 Cocoa 下的 3D 渲染框架，它并不支持古老的Objective-C语言[判断题]

A. 正确

B. 错误

23. 对 Sprite Kit 来说，Scean Kit中的3D 模型可以与 2D 精灵混合使用吗？[判断题]

A. 可以

B. 不可以

24. 对 Scene Kit 来说，Sprite Kit 中的场景和纹理可以作为 Scene Kit 的纹理贴图吗？[判断题]

A. 可以

B. 不可以

25. SceneKit 中的坐标系是右手坐标系，即笛卡尔坐标系吗？[判断题]

A. 是的
B. 不是

26. 由于Metal非常底层，所以它允许你使硬件达到运行效率的峰值，对你的游戏如何运行有着完全的控制？[判断题]

A. 正确
B. 错误

27. 在线播放视频一般访问服务器中的哪种类型文件？[单选题]

A. M3U8
B. flv
C. MP4
D. data

28. 以下UIView动画曲线类型，哪种表示动画速度由慢变快？[单选题]

A. easeInOut
B. easeIn
C. easeOut
D. linear

29. Quartz 2D不具有以下哪些功能？[单选题]

A. path-based drawing
B. offscreen rendering
C. Animation
D. anti-aliased rendering

30. 关于AVPlayerLayer实例的videoGravity属性，以下哪一项可以保留视频的宽高比，并对视频进行缩放，以填满层的范围区域？[单选题]

A. AVLayerVideoGravityResizeAspectFill
B. AVLayerVideoGravityResizeAspect
C. AVLayerVideoGravityResize
D. 都不是

31. 以下哪种框架可以让开发者自由、方便地调整全部或部分文字的颜色、尺寸、位置布局等属性，可以更加便捷地创建杂志、日记类的应用。[单选题]

A. UIKit
B. Foundation
C. Core Text
D. Core Image

32. 以下哪个是Core Text的字形绘制的最小单位？[单选题]

A. CTFrame
B. CTRun
C. CTLine
D. CTPath

33. Core Image图像处理框架是从哪个iOS 版本起加入进来的？[单选题]

A. iOS 5
B. iOS 6
C. iOS 7
D. iOS 8

34. 图像处理框架是基于什么对图像进行分析与处理的？[单选题]

A. 颜色
B. 像素
C. 点
D. 直方图

35. Core Image在iOS 6之后的滤镜数量增加至多少个？[单选题]

A. 24
B. 36
C. 48
D. 93

36. 自哪个版本的iOS开始，Apple为用户带来了炫酷的毛玻璃效果？[单选题]

A. iOS 6.0
B. iOS 7.0
C. iOS 8.0
D. iOS 9.0

37. CGImageRef、CGColorRef 两种数据类型是定义在哪个框架中的？[单选题]

A. Foundation
B. CoreGraphics
C. UIKit
D. Core Image

38. CAMetalLayer并不在Metal框架中，而是在以下哪个框架中？[单选题]

A. UIKit
B. QuartzCore
C. Foundation
D. Core Image

39. 以下哪些Cocoa Touch框架，用于处理音频和视频：[多选题]

A. Core Audio
B. OpenAL
C. Media Library
D. AV Foundation

40. 以下哪些Cocoa Touch框架，用于图形和动画：[多选题]

A. Metal
B. Core Animation
C. OpenGL ES
D. Quartz 2D

41. iOS中的动画实现技术主要是哪两个框架？[多选题]

A. Core Animation
B. Core Graphic
C. Foundation
D. UIKit

42. 以下UIView的哪些属性可以产生平滑的动画？[多选题]

A. frame
B. bounds
C. center
D. backgroundColor

43. 在iOS开发中，可以使用哪些音频播放技术方案？[多选题]

A. System Sound Services
B. AVAudioPlayer
C. Audio Queue Services
D. Open AL

44. 使用System Sound Services播放音频，有哪些特点？[多选题]

A. 播放的声音长度要小于30秒
B. 声音文件的格式必须是PCM或IMA4（IMA/ADPCM）
C. 可以控制音频播放的进度
D. 调用方法后立即播放声音，并且无法设置声音大小

45. 使用AVAudioPlayer技术，可以实现以下哪些功能？[多选题]

A. 播放任意时长的音频文件
B. 播放文件中或者内存缓存区中的声音
C. 进行音频文件的循环播放
D. 使用多个AVAudioPlayer实例，可以同时播放多个音频文件

46. 在iOS开发中播放视频，通常采用哪两种方式？[多选题]

A. OpenGL ES
B. Core Graphics
C. MPMoviePlayerController
D. AVPlayer

47. Quartz 2D 在 iOS 开发中，可以完成以下哪些任务？[多选题]

A. 绘制一些系统 UIKit 框架中不好展示的内容，例如饼图
B. 绘制图形：线条、三角形、矩形、圆、弧等
C. 读取和生成 PDF
D. 截图和裁剪图片

48. 以下关于TextKit，说法正确的是：[多选题]

A. 为了给开发者提供更高效的文字排版工具，在iOS 7.0中Apple向开发者推出了Text Kit类库。
B. TextKit类库是建立在Core Text基础之上的。
C. TextKit和UILabel、UITextView等控件紧密结合，在使用这些控件时，可以直接使用Textkit相关的功能。
D. Text Kit和Web Kit一样，都是建立在Core Graphics和Core Text框架之上。

49. 使用Core Image可以完成下面哪些工作？[多选题]

A. 使用内置的滤镜对图片进行快速的艺术处理和加工
B. 对人脸等特征进行检测
C. 同时使用多个滤镜以产生更加复杂多变的自定义效果
D. 创建运行在GPU上的自定义滤镜提高图像处理的速度

50. Core Image可以操作哪些来源的图像数据？[多选题]

A. Core Graphics
B. Core Data
C. Core Video
D. Image I/O

51. 使用Core Image框架的哪些技术，可以实现人脸的检测？[多选题]

A. CIFilter
B. CIDetector
C. CIFeature
D. CIFrame

52. 以下关于CADisplayLink和Timer，描述正确的是：[多选题]

A. CADisplayLink 使用场合相对专一，适合做 UI 的不停重绘。
B. NSTimer 的使用范围要广泛的多，各种需要单次或者循环定时处理的任务都可以使用。
C. 在UI相关的动画或者显示内容使用 CADisplayLink 比起用 NSTimer 的好处就是我们不需要再格外关心屏幕的刷新频率。
D. CADisplayLink可用于自定义动画引擎或者视频播放的渲染。

53. 关于Core Animation动画和UIView 动画，以下说法正确的是：[多选题]

A. Core Animation动画一切都是假象，并不会真实的改变图层的属性。
B. 如果在播放动画的时候，不需要与用户交互，推荐使用Core Animation动画。
C. UIView 动画一切都是假象，并不会真实的改变图层的属性。
D. UIView 动画必须通过修改属性的真实值，才有动画效果。

54. ARKIt主要有哪三层核心技术？[多选题]

A. 快速稳定的世界定位 ，包括实时运算，运动定位
B. 平面和边界感知 碰撞测试和光线估算，让虚拟内容和现实环境无缝衔接
C. 支持各种渲染制作工具
D. 支持各种三维模型制作工具

55. iOS 7带来的Sprite Kit框架支持以下哪些内容？[多选题]

A. 精灵
B. 场景
C. 特效
D. 物理库

56. Scene Kit支持以下哪些特效技术？[多选题]

A. 粒子效果
B. 物理引擎
C. 脚本事件
D. 多通道分层渲染

57. Scene Kit 提供了哪些类型的光照：[多选题]

A. 环境光
B. 定向光源
C. 点光源
D. 聚光灯

58. 第一个渲染 pass 永远是 Scene Kit 的默认渲染，它输出场景的哪些元素？[多选题]

A. 环境光
B. 粒子效果
C. 颜色
D. 景深

59. 在iOS 8里，苹果发布了一个新的接口叫做Metal，以下对Metal描述正确的是：[多选题]

A. Metal和OpenGL ES相似，它也是一个底层API。
B. Metal负责和3D绘图硬件交互。
C. Metal不是跨平台的。
D. Metal是一个支持GPU加速3D绘图的API。

1 : A	2 : A	3 : A	4 : B	5 : A	6 : B	7 : B	8 : A
9 : A	10 : A	11 : B	12 : A	13 : B	14 : A	15 : B	16 : A
17 : B	18 : B	19 : A	20 : B	21 : A	22 : B	23 : A	24 : A
25 : A	26 : A	27 : A	28 : B	29 : C	30 : A	31 : C	32 : B
33 : A	34 : B	35 : D	36 : B	37 : B	38 : B	39 : A,B,C,D	40 : A,B,C,D
41 : A,D	42 : A,B,C,D	43 : A,B,C,D	44 : A,B,D	45 : A,B,C,D	46 : C,D	47 : A,B,C,D	48 : A,B,C,D
49 : A,B,C,D	50 : A,C,D	51 : B,C	52 : A,B,C,D	53 : A,B,D	54 : A,B,C	55 : A,B,C,D	56 : A,B,C,D
57 : A,B,C,D	58 : C,D	59 : A,B,C,D					

十.安全篇

共有27题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 3DES和SHA1属于非对称加密的技术？[判断题]

A. 正确

B. 错误

2. MD5属于非对称加密的技术？[判断题]

A. 正确

B. 错误

3. 3DES是三重数据加密算法的通称，它相当于对每个数据块应用三次DES加密算法？[判断题]

A. 正确

B. 错误

4. 3DES通过增加DES的密钥长度来避免类似的攻击，而不是设计一种全新的块密码算法。[判断题]

A. 正确

B. 错误

5. 不可以从消息摘要中复原信息，并且两个不同的消息不会产生同样的消息摘要，这样描述SHA1加密算法正确吗？[判断题]

A. 正确

B. 错误

6. MD5加密算法是可逆的？[判断题]

A. 正确

B. 错误

7. Library目录下除了 Caches 子目录外，其它子目录及其文件都可以通过 iTunes 进行备份？[判断题]

A. 正确

B. 错误

8. 沙盒中的tmp目录的内容不会通过 iTunes 备份，程序员也不需要管 tmp 文件夹中的释放？[判断题]

A. 正确

B. 错误

9. Keychain是独立于每个App的沙盒之外的，所以即使App被删除之后，Keychain里面的信息依然存在？[判断题]

A. 不是
B. 是

10. 在沙盒环境测试appStore内购流程的时候，可以使用越狱的设备吗？[判断题]

A. 可以
B. 不可以

11. 在沙盒环境测试appStore内购流程的时候，项目Bundle identifier如果和申请AppID时填写的bundleID不一致，可以请求到内购项目吗？[判断题]

A. 可以
B. 不可以

12. 在沙盒环境测试appStore内购流程的时候，必须使用测试账号，而无法使用真实的Apple账号吗？[判断题]

A. 正确
B. 错误

13. 对于长度小于2^64位的消息，SHA1会产生一个多少位的消息摘要？[单选题]

A. 32
B. 64
C. 128
D. 160

14. MD5可以将任意长度的字符串变换成一个多少bit的大整数？[单选题]

A. 32
B. 64
C. 128
D. 256

15. UserDefaults 存储的数据，位于沙盒中的哪个目录？[单选题]

A. Caches/
B. Documents/
C. Library/Preferences
D. Temp/

16. 当iPhone重启时，会丢弃哪个沙盒目录下的文件？[单选题]

A. Documents
B. Library
C. Preferences
D. tmp

17. 沙盒下的哪个目录下的所有文件，都可以通过 iTunes 进行备份和恢复？[单选题]

A. Documents
B. Library
C. Caches
D. tmp

18. 沙盒下的哪个目录用来存放缓存文件，保存从网络下载的请求数据，例如网络下载的离线数据，图片，视频文件等？[单选题]

A. Documents
B. Library/Preferences
C. Library/Caches
D. tmp

19. 应用程序 bundle 的配置信息，包括了应用程序唯一标识名、版本号、可执行文件名等信息。这些信息都存储在哪个文件中？[单选题]

A. config.plist
B. property.plist
C. info.plist
D. userInfo.plist

20. 在info.plist中，以下哪个键用来存储应用程序的唯一标识？[单选题]

A. CFBundleDisplayName
B. CFBundleExecutable
C. CFBundleVersion
D. CFBundleIdentifier

21. 以下哪种操作可以有效地保护应用程序中的数据？[多选题]

A. 对设备中存储的内容进行加密。
B. 对网络传输的数据进行加密。
C. 使用post请求传递网络数据。
D. 对账户密码进行加盐操作。

22. 以下关于DES加密，描述正确的是：[多选题]

A. 提供高质量的数据保护，防止数据未经授权的泄露和未被察觉的修改
B. 具有较高的复杂性，使得破译的开销超过可能获得的利益，同时又要便于理解和掌握
C. 密码体制的安全性应该不依赖于算法的保密，其安全性仅以加密密钥的保密为基础
D. 实现经济，运行有效，并且适用于多种完全不同的应用

23. CommonCryptor.h提供了以下哪些加密算法：[多选题]

A. kCCAlgorithmAES128
B. kCCAlgorithmDES
C. kCCAlgorithm3DES
D. kCCAlgorithmBlowfish

24. 苹果的iOS系统采用了以下哪些严格的安全机制？[多选题]

A. 代码签名
B. 权限隔离
C. 可信启动链
D. 沙盒执行环境

25. 以下关于归档描述正确的是：[多选题]

A. 归档是一个过程，即用某种格式来保存一个或多个对象，以便以后还原这些对象。

B. 可以使用归档的方法进行对象的深复制。

C. 采用归档的形式来保存数据，该数据对象需要遵守 NSCoding 协议，并且必须提供 encodeWithCoder: 和 initWithCoder: 方法。

D. 归档的缺点是如果想改动归档数据的某一小部分，则需要解压整个数据和归档整个数据。

26. 应用程序 bundle 中主要有以下哪几种类型的文件？[多选题]

A. temp临时文件：用来存储程序运行期间的缓存文件。

B. 可执行文件：每个应用程序必须要有一个可执行文件。

C. 资源文件：是可执行文件以外的数据文件，常用的如图像、图标、音频文件、视图文件、配置文件等。

D. Info.plist：用来配置应用的基本参数信息。包括版本号，指向的可执行文件名、包名等。

27. 以下关于苹果对沙盒的限制，描述正确的是：[多选题]

A. 应用程序在自己的沙盒中运作，但是不能访问任何其他应用程序的沙盒；

B. 应用之间不能共享数据，沙盒里的文件不能被复制到其他沙盒；

C. 应用程序的文件夹中，也不能把其他应用文件夹复制到沙盒中；

D. 苹果禁止任何读写沙盒以外的文件，禁止应用程序将内容写到沙盒以外的文件夹中；

1 : B	2 : A	3 : A	4 : A	5 : A	6 : B	7 : A	8 : A
9 : B	10 : B	11 : B	12 : A	13 : D	14 : C	15 : C	16 : D
17 : A	18 : C	19 : C	20 : D	21 : A,B,D	22 : A,B,C,D	23 : A,B,C,D	24 : A,B,C,D
25 : A,B,C,D	26 : B,C,D	27 : A,B,C,D					

面试题第一篇

共有80题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

1. 被id声明的对象有什么特性?

参考答案：Id 声明的对象具有运行时的特性，即可以指向任意类型的objcetive-c的对象。

2. Objective-C有哪些内存管理方式?

参考答案：Objective-C的内存管理主要有三种方式ARC(自动内存计数)、手动内存计数、内存池。

3. 常见的object-c的数据类型有那些， 和C的基本数据类型有什么区别?

参考答案：object-c的数据类型有NSString，NSNumber，NSArray， NSMutableArray，NSData等等，这些都是class，创建后便是对象，而C语言的基本数据类型int，只是一定字节的内存空间，用于存放数值;NSInteger是基本数据类型，并不是NSNumber的子类，当然也不是NSObject的子类。

NSInteger是基本数据类型Int或者Long的别名(NSInteger的定义typedef long NSInteger)，它的区别在于，NSInteger会根据系统是32位还是64位来决定是本身是int还是Long。

4. 对于语句NSString*obj = [[NSData alloc] init]; obj在编译时和运行时分别是什么类型的对象?

参考答案：编译时是NSString的类型;运行时是NSData类型的对象。

5. 写一个setter方法用于完成@property(nonatomic,retain)NSString *name 。写一个setter方法用于完成@property(nonatomic，copy)NSString *name

参考答案：

```
- (void) setName:(NSString*) str
{
	[str retain];
	[name release];
	name = str;
}
- (void)setName:(NSString *)str
{
	id t = [str copy];
	[name release];
	name = t;
}
```

6. readwrite，readonly，assign，retain，copy，nonatomic各是有什么作用?

参考答案：1. readwrite 是可读可写特性;

2. readonly是只读特性，只会生成getter方法，不会生成setter方法 ;不希望属性在类外改变

3. assign 是赋值特性，setter方法将传入参数赋值给实例变量;仅设置变量时;

4. retain 表示持有特性，setter方法将传入参数先保留，再赋值，传入参数的retaincount会+1;

5. copy 表示赋值特性，setter方法将传入对象复制一份;需要完全一份新的变量时。

6. nonatomic 非原子操作，决定编译器生成的setter getter是否是原子操作，atomic表示多线程安全，一般使用nonatomic

7. #import 跟#include 又什么区别，@class呢, #import<> 跟 #import ""又什么区别?

参考答案：#import是Objective-C导入头文件的关键字，#include是C/C++导入头文件的关键字。

使用#import头文件会自动只导入一次，不会重复导入，相当于#include和#pragma once;

@class则告诉编译器某个类的声明，当执行时，才去查看类的实现文件，可以解决头文件的相互包含;

#import ""用来包含系统的头文件，#import用来包含用户头文件。

8. 原子(atomic)和非原子(non-atomic)属性有什么区别?

参考答案：1. atomic提供多线程安全。是防止在写未完成的时候被另外一个线程读取，造成数据错误。

2. non-atomic:在自己管理内存的环境中，解析的访问器保留并自动释放返回的值，如果指定了nonatomic ，那么访问器只是简单地返回这个值。

9. Object C中创建线程的方法是什么?

参考答案：线程创建有三种方法：使用NSThread创建、使用GCD的dispatch、使用子类化的NSOperation,然后将其加入NSOperationQueue;

10. 如果在主线程中执行代码，方法是什么?

参考答案：在主线程执行代码，方法是performSelectorOnMainThread

11. 如果想延时执行代码、方法是什么?

参考答案：如果想延时执行代码可以用performSelector:onThread:withObject:waitUntilDone:

12. 描述一下iOS SDK中如何实现MVC的开发模式

参考答案：MVC是模型、试图、控制开发模式，对于iOS SDK，所有的View都是视图层的，它应该独立于模型层，由视图控制层来控制。

所有的用户数据都是模型层，它应该独立于视图。

所有的ViewController都是控制层，由它负责控制视图，访问模型数据。

13. Objective-C类别的作用?继承和类别在实现中有何区别?

参考答案：category可以在不获悉，不改变原来代码的情况下往里面添加新的方法，只能添加，不能删除修改。并且如果类别和原来类中的方法产生名称冲突，则类别将覆盖原来的方法，因为类别具有更高的优先级。

类别主要有3个作用：
(1)将类的实现分散到多个不同文件或多个不同框架中。
(2)创建对私有方法的前向引用。
(3)向对象添加非正式协议。

继承可以增加，修改或者删除方法，并且可以增加属性。

14. Object-c的类可以多重继承么?可以实现多个接口么?Category是什么?重写一个类的方式用继承好还是分类好?为什么?

参考答案：Object-c的类不可以多重继承;
可以实现多个接口，通过实现多个接口可以完成C++的多重继承;
Category是类别，一般情况用分类好，用Category去重写类的方法，仅对本Category有效，不会影响到其他类与原有类的关系。

15. 写一个NSString类的实现

参考答案：+ (id)initWithCString:(c*****t char *)nullTerminatedCString encoding:(NSStringEncoding)encoding;

```
+ (id)stringWithCString:(c*****t char*)nullTerminatedCString encoding:(NSStringEncoding)encoding
{
    NSString *obj;
    obj = [self allocWithZone: NSDefaultMallocZone()];
    obj = [obj initWithCString: nullTerminatedCString encoding: encoding];
    return AUTORELEASE(obj);
}
```

16. static 关键字的作用

参考答案：1、函数体内 static 变量的作用范围为该函数体，不同于 auto 变量，该变量的内存只被分配一次， 因此其值在下次调用时仍维持上次的值；
2、在模块内的 static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；
3、在模块内的 static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明 它的模块内；
4、在类中的 static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；
5、在类中的 static 成员函数属于整个类所拥有，这个函数不接收 this 指针，因而只能访问类的static 成员变量

17. 线程与进程的区别和联系

参考答案：进程和线程都是由操作系统所体会的程序运行的基本单元，系统利用该基本单元实现系统对应用的并发性。

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。

进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。

线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。

18. 堆和栈的区别

参考答案：管理方式：对于栈来讲，是由编译器自动管理，无需我们手工控制；对于堆来说，释放工作由程序员控制，容易产生memory leak。
申请大小-栈：在 WINDOWS下，栈的大小是2M（也有的说是1M，总之是一个编译时就确定的常数），如果申请的空间超过栈的剩余空间时，将提示overflow。
申请大小-堆：堆是向高地址扩展的数据结构，是不连续的内存区域。这是由于系统是用链表来存储的空闲内存地址的，自然是不连续的，而链表的遍历方向是由低地址向高地址。堆的大小受限于计算机系统中有效的虚拟内存。由此可见，堆获得的空间比较灵活，也比较大。
碎片问题：对于堆来讲，频繁的new/delete势必会造成内存空间的不连续，从而造成大量的碎片，使程序效率降低。对于栈来讲，则不会存在这个问题，因为栈是先进后出的队列，他们是如此的一一对应，以至于永远都不可能有一个内存块从栈中间弹出
分配方式：堆都是动态分配的，没有静态分配的堆。栈有2种分配方式：静态分配和动态分配。静态分配是编译器完成的，比如局部变量的分配。动态分配由alloca函数进行分配，但是栈的动态分配和堆是不同的，他的动态分配是由编译器进行释放，无需我们手工实现。
分配效率：栈是机器系统提供的数据结构，计算机会在底层对栈提供支持：分配专门的寄存器存放栈的地址，压栈出栈都有专门的指令执行，这就决定了栈的效率比较高。堆则是C/C++函数库提供的，它的机制是很复杂的。申请大小-栈：在Windows下,栈是向低地址扩展的数据结构，是一块连续的内存的区域。这句话的意思是栈顶的地址和栈的最大容量是系统预先规定好的，在 WINDOWS下，栈的大小是2M（也有的说是1M，总之是一个编译时就确定的常数），如果申请的空间超过栈的剩余空间时，将提示overflow。因此，能从栈获得的空间较小。
申请大小-堆：堆是向高地址扩展的数据结构，是不连续的内存区域。这是由于系统是用链表来存储的空闲内存地址的，自然是不连续的，而链表的遍历方向是由低地址向高地址。堆的大小受限于计算机系统中有效的虚拟内存。由此可见，堆获得的空间比较灵活，也比较大。
碎片问题：对于堆来讲，频繁的new/delete势必会造成内存空间的不连续，从而造成大量的碎片，使程序效率降低。对于栈来讲，则不会存在这个问题，因为栈是先进后出的队列，他们是如此的一一对应，以至于永远都不可能有一个内存块从栈中间弹出
分配方式：堆都是动态分配的，没有静态分配的堆。栈有2种分配方式：静态分配和动态分配。静态分配是编译器完成的，比如局部变量的分配。动态分配由alloca函数进行分配，但是栈的动态分配和堆是不同的，他的动态分配是由编译器进行释放，无需我们手工实现。
分配效率：栈是机器系统提供的数据结构，计算机会在底层对栈提供支持：分配专门的寄存器存放栈的地址，压栈出栈都有专门的指令执行，这就决定了栈的效率比较高。堆则是C/C++函数库提供的，它的机制是很复杂的。

19. 什么是键-值,键路径是什么

参考答案：模型的性质是通过一个简单的键（通常是个字符串）来指定的。视图和控制器通过键来查找相应的属性值。

在一个给定的实体中，同一个属性的所有值具有相同的数据类型。键-值编码技术用于进行这样的查找—它是一种间接访问对象属性的机制。

键路径是一个由用点作分隔符的键组成的字符串，用于指定一个连接在一起的对象性质序列。第一个键的性质是由先前的性质决定的，接下来每个键的值也是相对于其前面的性质。

键路径使您可以以独立于模型实现的方式指定相关对象的性质。通过键路径，您可以指定对象图中的一个任意深度的路径，使其指向相关对象的特定属性

20. 如何理解Target-action?

参考答案：Target-action是一种设计模式，直译过来就是目标-动作机制。

Target目标即我们要调用哪个目标对象中的方法。Action动作是行为的意思，就是我们要调用目标对象中的哪一个方法。

当事件发生时，事件会被发送到控件对象中，然后再由这个控件对象去触发target对象上的action行为，来最终处理事件。

因此，Target-Action机制由两部分组成：即目标对象和行为Selector。目标对象指定最终处理事件的对象，而行为Selector则是处理事件的方法。

21. 请谈谈你所理解的objc的内存管理。

参考答案：如果您通过分配和初始化（比如[[MyClass alloc] init]）的方式来创建对象，您就拥有这个对象，需要负责该对象的释放。这个规则在使用NSObject的便利方法new 时也同样适用。

如果您拷贝一个对象，您也拥有拷贝得到的对象，需要负责该对象的释放。

如果您保持一个对象，您就部分拥有这个对象，需要在不再使用时释放该对象。

反过来，如果您从其它对象那里接收到一个对象，则您不拥有该对象，也不应该释放它（这个规则有少数的例外，在参考文档中有显式的说明）

22. 自动释放池是什么,如何工作的?

参考答案：当您向一个对象发送一个autorelease消息时，Cocoa就会将该对象的一个引用放入到最新的自动释放池。 它仍然是个正当的对象，因此自动释放池定义的作用域内的其它对象可以向它发送消息。
当程序执行到作用域结束的位置时，自动释放池就会被释放，池中的所有对象也就被释放。

1. ojc-c 是通过一种referring counting(引用计数)的方式来管理内存的, 对象在开始分配内存(alloc)的时候引用计数为一,以后每当碰到有copy,retain的时候引用计数都会加一, 每当碰到release和autorelease的时候引用计数就会减一,如果此对象的计数变为了0, 就会被系统销毁.

2. NSAutoreleasePool 就是用来做引用计数的管理工作的,这个东西一般不用你管的.

3. autorelease和release没什么区别,只是引用计数减一的时机不同而已,autorelease会在对象的使用真正结束的时候才做引用计数减一.

23. 类工厂方法是什么?
参考答案：类工厂方法的实现是为了向客户提供方便，它们将分配和初始化合在一个步骤中，返回被创建的对象，并进行自动释放处理。这些方法的形式是+ (type)className...（其中 className不包括任何前缀）。

工厂方法可能不仅仅为了方便使用。它们不但可以将分配和初始化合在一起，还可以为初始化过程提供对象的分配信息。

类工厂方法的另一个目的是使类提供单例对象。虽然init...方法可以确认一个类在每次程序运行过程只存在一个实例，但它需要首先分配一个新的实例，然后还必须释放该实例。工厂方法则可以避免为可能没有用的对象盲目分配内存。

24. 单件实例是什么?
参考答案：Foundation 和 Application Kit 框架中的一些类只允许创建单件对象，即这些类在当前进程中的唯一实例。

举例来说，NSFileManager 和NSWorkspace 类在使用时都是基于进程进行单件对象的实例化。当向这些类请求实例的时候，它们会向您传递单一实例的一个引用，如果该实例还不存在，则首先进行实例的分配和初始化。单件对象充当控制中心的角色，负责指引或协调类的各种服务。

如果类在概念上只有一个实例（比如NSWorkspace），就应该产生一个单件实例，而不是多个实例；如果将来可能有多个实例，您可以使用单件实例机制，而不是工厂方法或函数。

25. 什么是动态绑定?
参考答案：在运行时确定要调用的方法，动态绑定将调用方法的确定推迟到运行时。在编译时，方法的调用并不和代码绑定在一起，只有在消实发送出来之后，才确定被调用的代码。通过动态类型和动态绑定技术，您的代码每次执行都可以得到不同的结果。

运行时因子负责确定消息的接收者和被调用的方法。运行时的消息分发机制为动态绑定提供支持。当您向一个动态类型确定了的对象发送消息时，运行环境系统会通过接收者的isa指针定位对象的类，并以此为起点确定被调用的方法，方法和消息是动态绑定的。

而且，您不必在Objective-C 代码中做任何工作，就可以自动获取动态绑定的好处。您在每次发送消息时，特别是当消息的接收者是动态类型已经确定的对象时，动态绑定就会例行而透明地发生。

26. 请描述下Objective-C有哪些优缺点?
参考答案：Objective-C的优点：1、Cateogies；2、Posing；3、动态识别；4、指标计算；5、弹性讯息传递；6、不是一个过度复杂的C 衍生语言；7、Objective-C与C++可混合编程

缺点: 1、不支援命名空间；2、不支持运算符重载；3、不支持多重继承；4、使用动态运行时类型，所有的方法都是函数调用，所以很多编译时优化方法都用不到。

27. sprintf,strcpy,memcpy使用上有什么要注意的地方?
参考答案：strcpy是一个字符串拷贝的函数，它的函数原型为strcpy(char *dst, c*****t char *src); 将src开始的一段字符串拷贝到dst开始的内存中去，结束的标志符号为'\0'，由于拷贝的长度不是由我们自己控制的，所以这个字符串拷贝很容易出错。

具备字符串拷贝功能的函数有memcpy，这是一个内存拷贝函数，它的函数原型为memcpy(char *dst, c*****t char* src, unsigned int len);将长度为len的一段内存，从src拷贝到dst中去，这个函数的长度可控。但是会有内存叠加的问题。

sprintf是格式化函数。将一段数据通过特定的格式，格式化到一个字符串缓冲区中去。sprintf格式化的函数的长度不可控，有可能格式化后的字符串会超出缓冲区的大小，造成溢出。

28. 什么时候用delegate，什么时候用Notification?

参考答案： delegate针对one-to-one关系，并且reciever可以返回值给sender，notification可以针对one-to-one/many/none,reciever无法返回值给sender.所以，delegate用于sender希望接受到reciever的某个功能反馈值，notification用于通知多个object某个事件。

29. 什么是KVC和KVO?

参考答案：KVC(Key-Value-Coding)内部的实现：一个对象在调用setValue的时候，

（1）首先根据方法名找到运行方法的时候所需要的环境参数。

（2）他会从自己isa指针结合环境参数，找到具体的方法实现的接口。

（3）再直接查找得来的具体的方法实现。

KVO（Key-Value- Observing）：当观察者为一个对象的属性进行了注册，被观察对象的isa指针被修改的时候，isa指针就会指向一个中间类，而不是真实的类。所以 isa指针其实不需要指向实例对象真实的类。所以我们的程序最好不要依赖于isa指针。在调用类的方法的时候，最好要明确对象实例的类名

30. ViewController 的 loadView, viewDidLoad, viewDidUnload 分别是在什么时候调用的?

参考答案：viewDidLoad在view 从nib文件初始化时调用。

loadView在controller的view为nil时调用。此方法在编程实现view时调用,view 控制器默认会注册memory warning notification。

当view controller的任何view没有用的时候，viewDidUnload会被调用，在这里实现将retain的view release,如果是retain的IBOutlet view 属性则不要在这里release,IBOutlet会负责release 。

31. 类别的作用是什么?

参考答案：类别主要有3个作用：

(1)将类的实现分散到多个不同文件或多个不同框架中。

(2)创建对私有方法的前向引用。

(3)向对象添加非正式协议。

类别的局限性 有两方面局限性：

(1)无法向类中添加新的实例变量，类别没有位置容纳实例变量。

(2)名称冲突，即当类别中的方法与原始类方法名称冲突时，类别具有更高的优先级。类别方法将完全取代初始方法从而无法再使用初始方法。 无法添加实例变量的局限可以使用字典对象解决

32. 关键字volatile有什么含意?并给出三个不同的例子。

参考答案：一个定义为volatile的变量是说这变量可能会被意想不到地改变，这样，编译器就不会去假设这个变量的值了。精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。

下面是volatile变量的几个例子：

- 并行设备的硬件寄存器（如：状态寄存器）
- 一个中断服务子程序中会访问到的非自动变量(Non-automatic variables)
- 多线程应用中被几个任务共享的变量
- 一个参数既可以是const还可以是volatile吗？解释为什么。
- 一个指针可以是volatile 吗？解释为什么。

33. @synthesize是系统自动生成getter和setter属性声明吗?
参考答案：@dynamic 是开发者自已提供相应的属性声明。对于只读属性需要提供 setter，对于读写属性需要提供 setter 和getter。

@synthesize 意思是，除非开发人员已经做了，否则由编译器生成相应的代码，以满足属性声明。

@dynamic的意思是告诉编译器,属性的获取与赋值方法由用户自己实现, 不自动生成。

34. Difference between shallow copy and deep copy?浅复制和深复制的区别?
参考答案：浅层复制：只复制指向对象的指针，而不复制引用对象本身。
深层复制：复制引用对象本身。
意思就是说我有个A对象，复制一份后得到A_copy对象后，对于浅复制来说，A和A_copy指向的是同一个内存资源，复制的只不过是是一个指针，对象本身资源 还是只有一份，那如果我们对A_copy执行了修改操作,那么发现A引用的对象同样被修改，这其实违背了我们复制拷贝的一个思想。深复制就好理解了,内存中存在了两份独立对象本身。

35. What is advantage of categories? What is difference between implementing a category and inheritance?类别的作用？继承和类别在实现中有何区别?
参考答案：类别只是对现有的类进行完善其功能的扩展。
继承也可以实现这一目的，但是继承实际上从新创建了一个新的类，可以达到同种目的，但是类别更倾向于从方法上对现有的类做一些很简单的功能上的拓展，比较方便，类别不能增加新的实例变量，如果要引入新的实例变量，只能继承了。

类别主要有3个作用：
(1)将类的实现分散到多个不同文件或多个不同框架中。
(2)创建对私有方法的前向引用。
(3)向对象添加非正式协议。
继承可以增加，修改或者删除方法，并且可以增加属性。

36. Difference between categories and extensions?类别和类扩展的区别。
参考答案：category和extensions的不同在于后者可以添加属性。另外后者添加的方法是必须要实现的。
extensions可以认为是一个私有的Category。

37. Difference between protocol in objective c and interfaces in java?oc中的协议和java中的接口概念有何不同?
参考答案：OC中的代理有2层含义，官方定义为 formal和informal protocol。前者和Java接口一样。 informal protocol中的方法属于设计模式考虑范畴，不是必须实现的，但是如果有实现，就会改变类的属性。

其实关于正式协议，类别和非正式协议我很早前学习的时候大致看过，也写在了学习教程里 —非正式协议概念其实就是类别的另一种表达方式—这里有一些你可能希望实现的方法，你可以使用他们更好的完成工作。
这个意思是，这些是可选的。比如需要一个更好的方法，我们就会申明一个这样的类别去实现。然后你在后期可以直接使用这些更好的方法。 这么看，总觉得类别这玩意儿有点像协议的可选协议。

38. What is purpose of delegates? 代理的作用?
参考答案：代理的目的是改变或传递控制链。允许一个类在某些特定时刻通知到其他类，而不需要获取到那些类的指针。可以减少框架复杂度。 另外一点，代理可以理解为java中的回调监听机制的一种类似。

39. What are mutable and immutable types in Objective-C?
参考答案：可修改不可修改的集合类。这个我个人简单理解就是可动态添加修改和不可动态添加修改一样。

比如NSArray和NSMutableArray。前者在初始化后的内存控件就是固定不可变的，后者可以添加等，可以动态申请新的内存空间

40. When we call objective c is runtime language what does it mean? 我们说的oc是动态运行时语言是什么意思?
参考答案：多态。主要是将数据类型的确定由编译时，推迟到了运行时。 这个问题其实浅涉及到两个概念，运行时和多态。

简单来说，运行时机制使我们直到运行时才去决定一个对象的类别，以及调用该类别对象指定方法。

多态：不同对象以自己的方式响应相同的消息的能力叫做多态。意思就是假设生物类（life）都用有一个相同的方法-eat; 那人类属于生物，猪也属于生物，都继承了life后，实现各自的eat，但是调用是我们只需调用各自的eat方法。 也就是不同的对象以自己的方式响应了相同的消息（响应了eat这个选择器）。 因此也可以说，运行时机制是多态的基础。

41. What is push notification? 什么是推送消息?
参考答案：推送通知更是一种技术。
简单点就是客户端获取资源的一种手段。
普通情况下，都是客户端主动的pull。
推送则是服务器端主动push。

42. 谈谈对多态性的理解?
参考答案：多态简单的说就是对于不同对象响应同一个方法时做出的不同反应（不同对象以自己的方式响应相同的消息）。在OC中动态类型id是实现多态的一种方式，id是一个独特的数据类型，可以转换为任何数据类型。

43. What is responder chain?
参考答案：响应者： 对用户交互动作事件进行响应的对象。
响应者链：成为处理事件的响应者的先后顺序链。

如果一个View有一个视图控制器（View Controller），它的下一个响应者是这个视图控制器，紧接着才是它的父视图（Super View），如果一直到Root View都没有处理这个事件，事件会传递到UIWindow（iOS中有一个单例Window），此时Window如果也没有处理事件，便进入UIApplication，UIApplication是一个响应者链的终点，它的下一个响应者指向nil，以结束整个循环。

44. Difference between frame and bounds? frame和bounds有什么不同?
参考答案：frame指的是：该view在父view坐标系统中的位置和大小。（参照点是父亲的坐标系统）

bounds指的是：该view在本身坐标系统中 的位置和大小。（参照点是本身坐标系统）

45. Difference between method and selector? 方法和选择器有何不同?
参考答案：selector是一个方法的名字，method是一个组合体，包含了名字和实现.

46. 什么是NSOperation queue?
参考答案：存放NSOperation的集合类。 操作和操作队列，基本可以看成java中的线程和线程池的概念。用于处理ios多线程开发的问题。

47. What is lazy loading?
参考答案：只在用到的时候才去初始化。 也可以理解成延时加载。

最好也最简单的一个列子就是tableView中图片的加载显示了。
一个延时载，避免内存过高，一个异步加载，避免线程堵塞。

48. Can we use two tableview controllers on one viewcontroller? 是否在一个视图控制器中嵌入两个tableview控制器?
参考答案：一个视图控制只提供了一个View视图，理论上一个tableViewController也不能放吧， 只能说可以嵌入一个tableview视图。当然一个视图控制器嵌入多少个table view都是可以的。

49. Can we use one tableview with two different datasources? How you will achieve this? 一个tableView是否可以关联两个不同的数据源？你会怎么处理?
参考答案：首先从代码来看，数据源如何关联上的，其实是在数据源关联的代理方法里实现的。
因此我们并不关心如何去关联他，他怎么关联上，方法只是让我返回根据自己的需要去设置如相关的数据源。
当然可以设置多个数据源，在数据源关联的代理方法合并两个数据源即可。

50. id、nil代表什么?
参考答案：id和void *并非完全一样。 id是指向struct objc_object的一个指针，这个意思基本上是说，id是一个指向任何一个继承了Object（或者NSObject）类的对象。需要注意的是id是一个指针，所以你在使用id的时候不需要加星号。
比如id foo=nil定义了一个nil指针，这个指针指向NSObject的一个任意子类。而id *foo=nil则定义了一个指针，这个指针指向另一个指针，被指向的这个指针指向NSObject的一个子类。

nil和C语言的NULL相同，在objc/objc.h中定义。
nil表示一个Objctive-C对象，这个对象的指针指向空（没有东西就是空）。
首字母大写的Nil和nil有一点不一样，Nil定义一个指向空的类（是Class，而不是对象）。
SEL是—selector‖的一个类型，表示一个方法的名字 Method（我们常说的方法）表示一种类型，这种类型与selector和实现(implementation)相关
IMP定义为 id (*IMP) (id, SEL, …)。

51. 层和UIView的区别是什么?
参考答案：两者最大的区别是,图层不会直接渲染到屏幕上，UIView是iOS系统中界面元素的基础，所有的界面元素都是继承自它。

它本身完全是由CoreAnimation来实现的。它真正的绘图部分，是由一个CALayer类来管理。

UIView本身更像是一个CALayer的管理器。一个UIView上可以有n个CALayer，每个layer显示一种东西，增强UIView的展现能力。

52. 你是如何理解GCD的?

参考答案：Grand Central Dispatch(GCD) 是Apple开发的一个多核编程的较新的解决方法。它主要用于优化应用程序以支持多核处理器以及其他对称多处理系统。它是一个在线程池模式的基础上执行的并行任务。在Mac OS X 10.6雪豹中首次推出，也可在IOS 4及以上版本使用。

它具有以下优点：
GCD 能通过推迟昂贵计算任务并在后台运行它们来改善应用的响应性能。
GCD 提供一个易于使用的并发模型而不仅仅只是锁和线程，以帮助我们避开并发陷阱。
GCD 具有在常见模式（例如单例）上用更高性能的原语优化你的代码的潜在能力。

53. 什么是可扩展标记语言extensible markup language XML？

参考答案：XML 指可扩展标记语言（EXtensible Markup Language）
XML 是一种标记语言，很类似 HTML
XML 的设计宗旨是传输数据，而非显示数据
XML 标签没有被预定义。您需要自行定义标签。
XML 被设计为具有自我描述性。
XML 是 W3C 的推荐标准

54. XML与HTML的区别

参考答案：XML与HTML的设计区别是：
XML的核心是数据，其重点是数据的内容。
而HTML 被设计用来显示数据，其重点是数据的显示。

XML和HTML语法区别：
HTML的标记不是所有的都需要成对出现，XML则要求所有的标记必须成对出现；
HTML标记不区分大小写，XML则大小敏感,即区分大小写。

XML 不是 HTML 的替代。
XML 和 HTML 为不同的目的而设计：
XML 被设计为传输和存储数据，其焦点是数据的内容。
HTML 被设计用来显示数据，其焦点是数据的外观。
HTML 旨在显示信息，而 XML 旨在传输信息。

55. HTTP协议的主要特点

参考答案：1.支持客户/服务器模式。

2.简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。

3.灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。

4.无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

5.无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

56. TCP/UDP区别联系

参考答案：TCP---传输控制协议,提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前，必须先在双方之间建立一个TCP连接，之后才能传输数据。TCP提供超时重发，丢弃重复数据，检验数据，流量控制等功能，保证数据能从一端传到另一端。

UDP---用户数据报协议，是一个简单的面向数据报的运输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快

57. socket连接和http连接的区别

参考答案：http连接：http连接就是所谓的短连接，即客户端向服务器端发送一次请求，服务器端响应后连接即会断掉；

socket连接：socket连接就是所谓的长连接，理论上客户端和服务器端一旦建立起连接将不会主动断掉； 但是由于各种环境因素可能会是连接断开，比如说：服务器端或客户端主机down了，网络故障，或者两者之间长时间没有数据传输，网络防火墙可能会断开该连接以释放网络资源。 所以当一个socket连接中没有数据的传输，那么为了维持连接需要发送心跳消息~~具体心跳消息格式是开发者自己定义的

58. 什么是TCP连接的三次握手

参考答案：第一次握手：客户端发送syn包(syn=j)到服务器，并进入SYN_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。

59. 利用Socket建立网络连接的步骤

参考答案：建立Socket连接至少需要一对套接字，其中一个运行于客户端，称为ClientSocket ，另一个运行于服务器端，称为ServerSocket。
套接字之间的连接过程分为三个步骤：服务器监听，客户端请求，连接确认。
1。服务器监听：服务器端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态，等待客户端的连接请求。
2。客户端请求：指客户端的套接字提出连接请求，要连接的目标是服务器端的套接字。为此，客户端的套接字必须首先描述它要连接的服务器的套接字，指出服务器端套接字的地址和端口号，然后就向服务器端套接字提出连接请求。
3。连接确认：当服务器端套接字监听到或者说接收到客户端套接字的连接请求时，就响应客户端套接字的请求，建立一个新的线程，把服务器端套接字的描述发给客户端，一旦客户端确认了此描述，双方就正式建立连接。而服务器端套接字继续处于监听状态，继续接收其他客户端套接字的连接请求

60. 堆和栈上的指针

参考答案：指针所指向的这块内存是在哪里分配的,在堆上称为堆上的指针,在栈上为栈上的指针.
在堆上的指针,可以保存在全局数据结构中,供不同函数使用访问同一块内存.
在栈上的指针,在函数退出后,该内存即不可访问.

61. 什么是指针的释放?

参考答案：具体来说包括两个概念.

1 释放该指针指向的内存,只有堆上的内存才需要我们手工释放,栈上不需要.

2 将该指针重定向为NULL.

62. 数据结构中的指针?

参考答案：其实就是指向一块内存的地址,通过指针传递,可实现复杂的内存访问.

63. 什么是函数指针?

参考答案：指向一块函数的入口地址.

64. 指针作为函数的参数?

参考答案：比如指向一个复杂数据结构的指针作为函数变量。这种方法避免整个复杂数据类型内存的压栈出栈操作,提高效率。 注意:指针本身不可变,但指针指向的数据结构可以改变.

65. 指向指针的指针?

参考答案：指针指向的变量是一个指针,即具体内容为一个指针的值,是一个地址. 此时指针指向的变量长度也是4位.

66. 指针与地址的区别?

参考答案：1、指针意味着已经有一个指针变量存在,他的值是一个地址,指针变量本身也存放在一个长度为四个字节的地址当中,而地址概念本身并不代表有任何变量存在.

2 指针的值,如果没有限制,通常是可以变化的,也可以指向另外一个地址. 地址表示内存空间的一个位置点,他是用来赋给指针的,地址本身是没有大小概念,指针指向变量的大小,取决于地址后面存放的变量类型.

67. 指针与数组名的关系?

参考答案：其值都是一个地址,但前者是可以移动的,后者是不可变的.

68. 怎样防止指针的越界使用问题?

参考答案：必须让指针指向一个有效的内存地址：

1 防止数组越界

2 防止向一块内存中拷贝过多的内容

3 防止使用空指针

4 防止改变const修改的指针

5 防止改变指向静态存储区的内容

6 防止两次释放一个指针

7 防止使用野指针.

69. 指针的类型转换?

参考答案：指针转换通常是指针类型和void * 类型之前进行强制转换,从而与期望或返回void指针的函数进行正确的交接.

70. static有什么用途？（请至少说明两种）

参考答案：1.限制变量的作用域

2.设置变量的存储域

71. 引用与指针有什么区别?

参考答案：1) 引用必须被初始化，指针不必。

2) 引用初始化以后不能被改变，指针可以改变所指的对象。

3) 不存在指向空值的引用，但是存在指向空值的指针。

72. 全局变量和局部变量在内存中是否有区别？如果有，是什么区别?

参考答案：全局变量储存在静态数据库，局部变量在堆栈

73. 什么是平衡二叉树?

参考答案：左右子树都是平衡二叉树且左右子树的深度差值的绝对值不大于1

74. 堆栈溢出一般是由什么原因导致的?

参考答案：没有回收垃圾资源

75. 什么函数不能声明为虚函数?

参考答案：constructor

76. 冒泡排序算法的时间复杂度是什么?

参考答案：O(n^2)

77. 写出float x 与—零值‖比较的if语句

参考答案：if(x>0.000001&&x<-0.000001)

78. Internet采用哪种网络协议？该协议的主要层次结构?

参考答案：tcp/ip
应用层
传输层
网络层
数据链路层
物理层

79. Internet物理地址和IP地址转换采用什么协议?

参考答案：ARP (Address Resolution Protocol)（地址解析協議）

80. IP地址的编码分为哪俩部分?

参考答案：IP地址由两部分组成，网络号和主机号。

面试题第二篇

共有80题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

81. 局部变量能否和全局变量重名?
参考答案：能，局部会屏蔽全局。要用全局变量，需要使用"::"
局部变量可以与全局变量同名，在函数内引用这个变量时，会用到同名的局部变量，而不会用到全局变量。

82. 如何引用一个已经定义过的全局变量?
参考答案：可以用引用头文件的方式，也可以用extern关键字。

83. 全局变量可不可以定义在可被多个.C文件包含的头文件中？为什么?
参考答案：可以，在不同的C文件中以static形式来声明同名全局变量。
可以在不同的C文件中声明同名的全局变量，前提是其中只能有一个C文件中对此变量赋初值，此时连接不会出错。

84. 语句for(；1；)有什么问题？它是什么意思?
参考答案：和while(1)相同。

85. do……while和while……do有什么区别?
参考答案：前一个循环一遍再判断，后一个判断以后再循环

86. Heap与stack的差别。
参考答案：Heap是堆，stack是栈。
Stack的空间由操作系统自动分配/释放，Heap上的空间手动分配/释放。
Stack空间有限，Heap是很大的自由存储区
C中的malloc函数分配的内存空间即在堆上,C++中对应的是new操作符。
程序在编译期对变量和函数分配内存都在栈上进行,且程序运行过程中函数调用时参数的传递也在栈上进行

87. void test() { char string[10]; char* str1 = "0123456789"; strcpy(string, str1); } 以上代码有什么问题?
参考答案：字符串str1需要11个字节才能存放下（包括末尾的‘\0‘），而string只有10个字节的空间，strcpy会导致数组越界

88. void test() { char string[10], str1[10]; int i; for(i=0; i<10; i++){str1 = 'a'; } strcpy(string, str1); } 代码有什么问题?
参考答案：字符数组str1不能在数组内结束；

strcpy(string, str1)调用使得从str1[url=]内存[/url]起复制到string内存起所复制的字节数具有不确定性。

89. void test(char* str1) { char string[10]; if(strlen(str1) <= 10) { strcpy(string, str1); } } 代码有什么问题?
参考答案：if(strlen(str1) <= 10)应改为if(strlen(str1) < 10)，因为strlen的结果未统计‘\0‘所占用的1个字节。

90. void GetMemory(char *p){p = (char *) malloc(100);}void Test(void){char *str = NULL; GetMemory(str);strcpy(str, "hello world");printf(str);}
参考答案：传入中GetMemory(char *p)函数的形参为字符串指针，在函数内部修改形参并不能真正的改变传入形参的值，执行完char *str = NULL; GetMemory(str);后的str仍然为NULL；

91. char *GetMemory(void) { char p[] = "hello world"; return p; } void Test(void) {char *str = NULL; str = GetMemory(); printf(str); }

参考答案：char p[] = "hello world";
return p;
p[]数组为函数内的局部自动变量，在函数返回后，内存已经被释放。
这是许多程序员常犯的错误，其根源在于不理解变量的生存期。

92. void GetMemory(char **p, int num) {*p = (char *) malloc(num);} void Test(void) {char *str = NULL; GetMemory(&str, 100); strcpy(str, "hello");printf(str);}

参考答案：传入GetMemory的参数为字符串指针的指针，但是在GetMemory中执行申请内存及赋值语句
*p = (char *) malloc(num);
后未判断内存是否申请成功，应加上：
if (*p == NULL)
{
...//进行申请内存失败处理
}

93. void Test(void) {char *str = (char *) malloc(100); strcpy(str, "hello"); free(str); ... //省略的其它语句 }

参考答案：在执行char *str = (char *) malloc(100);后未进行内存是否申请成功的判断；
另外，在free(str)后未置str为空，导致可能变成一个—野‖指针，应加上：str = NULL;

94. 分别给出BOOL，int，float，指针变量与—零值比较的 if 语句（假设变量名为var）

参考答案：BOOL型变量：if(!var)
int型变量： if(var==0)
float型变量：
const float EPSINON = 0.00001;
if ((x >= - EPSINON) && (x <= EPSINON)
指针变量： if(var==NULL)

95. Object-C 有多继承吗?没有的话用什么代替?

参考答案：Cocoa中所有的类都是NSObject的子类，多继承在这里是用protocol委托代理来实现的，你不用去考虑繁琐的多继承，虚基类的概念，多态性在Objective－C中通过委托来实现。

96. Object-C 有私有方法吗?私有变量呢?

参考答案：Objective－C类里面的方法只有两种，静态方法和实例方法，这似乎就不是完整的面向对象了,按照OO的原则就是一个对象只暴露有用的东西。

如果没有了私有方法的话，对于一些小范围的代码重用就不那么顺手可，在里面声明一个私有方法。

97. 关键字volatile有什么含义?并给出三个不同例子?

参考答案：一个定义为volatile的变量是说这变量可能会被意想不到地改变，这样，编译器就不会去假设这个变量的值了。

精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。

下面是volatile变量的几个例子：
• 并行设备的硬件寄存器（如：状态寄存器）
• 一个中断服务子程序中会访问到的非自动变量(Non-automatic variables)
• 多线程应用中被几个任务共享的变量

98. 一个参数既可以是const还可以是volatile吗？解释为什么。
参考答案：是的。
一个例子是只读的状态寄存器。
它是volatile因为它可能被意想不到地改变。
它是const因为程序不应该试图去修改它。

99. 一个指针可以是volatile吗？解释为什么。
参考答案：是的。
尽管这并不很常见。
一个例子是当一个中服务子程序修该一个指向一个buffer的指针时。

100. #import和#include的区别，@class 代表什么?
参考答案：@class一般用于头文件中需要声明该类的某个实例变量的时候用到，在m文件中还是需要使用#import
而#import比起#include的好处就是不会引起重复包含。

101. 请介绍一下Object-C 的内存管理方式
参考答案：1.当你使用new,alloc和copy方法创建一个对象时,该对象的保留计数器值为1.当你不再使用该对象时,你要负责向该对象发送一条release或autorelease消息.这样,该对象将在使用寿命结束时被销毁.

2.当你通过任何其他方法获得一个对象时,则假设该对象的保留计数器值为1,而且已经被设置为自动释放,你不需要执行任何操作来确保该对象被清理.如果你打算在一段时间内拥有该对象,则需要保留它并确保在操作完成时释放它.

3.如果你保留了某个对象,你需要(最终)释放或自动释放该对象.必须保持retain方法和release方法的使用次数相等.

102. 内存管理autorelease、retain、copy、assign 的set 方法和含义?
参考答案：你初始化(alloc/init)的对象，你需要释放(release)它。

例如：NSMutableArray aArray = [[NSArray alloc] init]; 后，需要[aArray release]; 你retain或copy的，你需要释放它。
例如：[aArray retain]后，需要[aArray release]; 被传递(assign)的对象，你需要斟酌的retain和release。
例如： obj2 = [[obj1 someMethod] autorelease];对象2接收对象1的一个自动释放的值，或传递一个基本数据类型(NSInteger，NSString)时： 你或希望将对象2进行retain，以防止它在被使用之前就被自动释放掉。但是在retain后，一定要在适当的时候进行释放。

关于索引计数(Reference Counting)的问题，retain值 = 索引计数(Reference Counting)。NSArray对象会retain(retain值加一)任何数组中的对象。当NSArray被卸载(dealloc)的时候，所有数组中的对象会被执行一次释放(retain值减一)。不仅仅是NSArray，任何收集类(Collection Classes)都执行类似操作。例如NSDictionary，甚至UINavigationController。 Alloc/init建立的对象，索引计数为1。无需将其再次retain。

103. 对象是什么时候被释放的?
参考答案：引用计数为0时。
autorelease实际上只是把对release的调用延迟了，对于每一个Autorelease，系统只是把该Object放入了当前的Autorelease pool中，当该pool被释放时，该pool中的所有Object会被调用Release。

对于每一个Runloop，系统会隐式创建一个Autorelease pool，这样所有的release pool会构成一个象CallStack一样的一个栈式结构，在每一个Runloop结束时，当前栈顶的Autorelease pool会被销毁，这样这个pool里的每个Object（就是autorelease的对象）会被release。

那什么是一个Runloop呢？一个UI事件，Timer call，delegate call，都会是一个新的Runloop。

104. 为什么很多内置的类，如TableViewController的delegate的属性是assign不是retain?

参考答案：循环引用所有的引用计数系统，都存在循环应用的问题。例如下面的引用关系：

• 对象a创建并引用到了对象b.

• 对象b创建并引用到了对象c.

• 对象c创建并引用到了对象b.

这时候b和c的引用计数分别是2和1。

当a不再使用b，调用release释放对b的所有权，因为c还引用了b，所以b的引用计数为1，b不会被释放。

b不释放，c的引用计数就是1，c也不会被释放。从此，b和c永远留在内存中。

这种情况，必须打断循环引用，通过其他规则来维护引用关系。

比如，我们常见的delegate往往是assign方式的属性而不是retain方式的属性，赋值不会增加引用计数，就是为了防止delegation两端产生不必要的循环引用。

如果一个UITableViewController 对象a通过retain获取了UITableView对象b的所有权， 这个UITableView对象b的delegate又是a， 如果这个delegate是retain方式的，那基本上就没有机会释放这两个对象了。

105. tableView的重用机制是什么?

参考答案：UITableView通过重用单元格来达到节省内存的目的:通过为每个单元格指定一个重用标识符(reuseIdentifier),即指定了单元格的种类,以及当单元格滚出屏幕时,允许恢复单元格以便重用.对于不同种类的单元格使用不同的ID,对于简单的表格,一个标识符就够了.

假如一个TableView中有10个单元格,但是屏幕上最多能显示4个,那么实际上iPhone只是为其分配了4个单元格的内存,没有分配10个,当滚动单元格时,屏幕内显示的单元格重复使用这4个内存。

106. ViewController 的 loadView 、 viewDidLoad 、 viewDidUnload 分别是什么时候调用的， 在自定义 ViewCointroller 时在这几个函数中应该做什么工作?

参考答案： loadView方法 当view需要被展示而它却是nil时， viewController会调用该方法。不要直接调用该方法。如果手工维护views，必须重载重写该方法。

viewDidLoad方法 此方法只有当view从nib文件初始化的时候才被调用。重载重写该方法以进一步定制view 在iPhone OS 3.0及之后的版本中，还应该重载重写viewDidUnload来释放对view的任何索引 viewDidLoad后调用数据Model viewDidUnload方法 当系统内存吃紧的时候会调用该方法

107. id、nil 代表什么?

参考答案：id和void *并非完全一样。 id是指向struct objc_object的一个指针，这个意思基本上是说，id是一个指向任何一个继承了Object（或者NSObject）类的对象。需要注意的是id是一个指针，所以你在使用id的时候不需要加星号。

比如id foo=nil定义了一个nil指针，这个指针指向NSObject的一个任意子类。而id *foo=nil则定义了一个指针，这个指针指向另一个指针，被指向的这个指针指向NSObject的一个子类。

nil nil和C语言的NULL相同，在objc/objc.h中定义。nil表示一个Objctive-C对象，这个对象的指针指向空（没有东西就是空）

108. 请简单介绍一下cocoa touch 框架

参考答案：iPhone OS 应用程序的基础 Cocoa Touch 框架重用了许多 Mac 系统的成熟模式，但是它更多地专注于触摸的接口和优化。UIKit 为您提供了在 iPhone OS 上实现图形，事件驱动程序的基本工具，其建立在和 Mac OS X 中一样的 Foundation 框架上，包括文件处理，网络，字符串操作等。

有了 UIKit，您可以使用 iPhone OS 上的独特的图形接口控件，按钮，以及全屏视图的功能，您还可以使用加速仪和多点触摸手势来控制您的应用。

Cocoa Touch 既包含只需要几行代码就可以完成全部任务的强大的 Objective-C 框架，也在需要时提供基础的 C 语言 API 来直接访问系统。

109. 自动释放池是什么,如何工作

参考答案：当您向一个对象发送一个autorelease消息时，Cocoa就会将该对象的一个引用放入到最新的自动释放池。它仍然是个正当的对象，因此自动释放池定义的作用域内的其它对象可以向它发送消息。当程序执行到作用域结束的位骆时，自动释放池就会被释放，池中的所有对象也就被释放。

1. ojc-c 是通过一种"referring counting"(引用计数)的方式来管理内存的, 对象在开始分配内存(alloc)的时候引用计数为一,以后每当碰到有copy,retain的时候引用计数都会加一, 每当碰到release和autorelease的时候引用计数就会减一,如果此对象的计数变为了0, 就会被系统销毁.

2. NSAutoreleasePool 就是用来做引用计数的管理工作的,这个东西一般不用你管的.3. autorelease和release没什么区别,只是引用计数减一的时机不同而已,autorelease会在对象的使用真正结束的时候才做引用计数减一.

110. 类工厂方法是什么

参考答案：类工厂方法的实现是为了向客户提供方便，它们将分配和初始化合在一个步骤中，返回被创建的对象，并进行自动释放处理。这些方法的形式是+ (type)className...（其中 className不包括任何前缀）。

工厂方法可能不仅仅为了方便使用。它们不但可以将分配和初始化合在一起，还可以为初始化过程提供对象的分配信息。

类工厂方法的另一个目的是使类（比如NSWorkspace）提供单件实例。虽然init...方法可以确认一个类在每次程序运行过程只存在一个实例，但它需要首先分配一个“生的”实例，然后还必须释放该实例。工厂方法则可以避免为可能没有用的对象盲目分配内存。

111. 程序在内存中运行时，内存分几个区？各自用途?

参考答案：栈区 由编译器自动分配释放存放函数的参数值，局部变量的值等。在高级语言中不需要显式的分配和释放

堆区 一般由程序员手动分配释放，如果不释放可有由OS释放。

数据区 存储全局和静态变量。初始化的全局和静态变量在一块区域，未初始化的放在相邻的一块区域，程序结束后由系统释放。

代码区 存放函数体的二进制代码。

112. 引用与指针有什么区别?

参考答案：引用是给变量起一个别名 也就是一个变量有两个名字 他们是同一块空间

指针是分配一块内存用来保存变量的地址 间接引用变量

113. int retVal = UIApplicationMain(argc, argv, nil, nil); 是什么意思?

参考答案：对UIApplication对象进行了初始化，这个方法除了argc 和 argv 参数外，另外这个函数还有2个两个字符串参数来识别UIApplication类和UIApplication代理类。
在这里默认是2个nil。
第一个参数为nil就默认把UIApplication类作为缺省值进行初始化，可以在这里不填nil而是使用自己定义的UIApplication子类。
至于第二个参数nil就设置为nil就把模板生成的HelloWorldAppdelegate类作为默认值。

114. 保存一个变量到本地，列举两个简单的方法。
参考答案：直接写入writeToFile 支持NSString NSArray NSDictionary

实现NSCoding NSCopy 协议

115. 如果UIView *view 已经实例化，在view添加了N个UIButton类的实例，并且button已经用tag区分开，如何找出其中指定的button改变它的某个属性?
参考答案：view中有一个方法可以根据tag值，把它上面的子视图找出来 [view subviewWithTag:];

116. 当A类中的某个方法执行到某处时， 这时想在B类中执行某个方法，如何做，举出一个方法说明?
参考答案：可以用委托方法来实现 在A类中初始化一个成员让他指向B类就可以调用B的方法了

117. 下面代码中obj2是否需要dealloc？ ClassA *obj1 = [[ClassA alloc] init];ClassA *obj2 = obj1;[obj1 hello];[obj1 dealloc];[obj2 hello];
参考答案：这个要根据hello的方法来定了 如果hello方法分配空间了那个释放再调用以后就会出错。

[obj2 dealloc];一般情况下我们不显式的调用dealloc方法 这个方法系统在对象引用计数为零的时候自动调用

另外obj2和obj1指向的是同一块空间释放一次就可以了

118. 类方法创造的对象要不要用release释放?
参考答案：任何方法创建的对象对遵从内存管理原则 用alloc方法分配的对象就需要释放

如果用类方法创建对象时候，没有用到alloc，那么分配空间的问题，已经在方法内部做了处理，所以我们就不需要去释放了

119. 想传输一张图片，可以用什么协议传输?
参考答案：可以用http ftp等协议。

120. 用NSURL初始化的NSData对象，用哪个类去解析?
参考答案：这个需要根据NSData数据是什么类型 然后可以用对应的解析工具来解析。

121. OC中加号方法与减号的区别?
参考答案：加号方法是类方法 可以用类名来调用

减号方法是实例方法 必须由类的实例来调用

122. 如何修饰私有变量?
参考答案：在OC中默认变量属性都是有私有的，我们可以property关键字来修饰，以便可以不同过接口直接访问。

123. 建一个工程用到最基本的两个框架是什么?
参考答案：cocoa Touch框架 Function框架

124. 一个UITableView的实例，重新加载数据的方法是什么?
参考答案：[tableView reloadData]方法

125. self.name = "object" name = "object"的区别
参考答案：self.name 需要要合成属性然后系统会默认生成set get方法 所以在赋值是会调用set方法

name = "object"则直接赋值

126. 怎么解决缓存池满的问题(cell)
参考答案：ios中不存在缓存池满的情况，因为通常我们ios中开发，对象都是在需要的时候才会创建，有种常用的说话叫做懒加载，还有在UITableView中一般只会创建刚开始出现在屏幕中的cell，之后都是从缓存池里取，不会在创建新对象。

缓存池里最多也就一两个对象，缓存池满的这种情况一般在开发java中比较常见，java中一般把最近最少使用的对象先释放。

127. 如何渲染自定义格式字符串的UILabel
参考答案：通过NSAttributedString类

128. 控制器View的生命周期及相关函数是什么？你在开发中是如何用的?
参考答案：1.首先判断控制器是否有视图，如果没有就调用loadView方法创建：通过storyboard或者代码；

2.随后调用viewDidLoad，可以进行下一步的初始化操作；只会被调用一次；

3.在视图显示之前调用viewWillAppear；该函数可以多次调用；

4.视图viewDidAppear

5.在视图显示之前调用viewWillDisappear；该函数可以多次调用；

6.在布局变化前后，调用viewWill/DidLayoutSubviews处理相关信息；

129. 请描述NSRunLoop的实现机制,及在多线程中如何使用。
参考答案：NSRunLoop是IOS消息机制的处理模式。

1.NSRunLoop的主要作用：控制NSRunLoop里面线程的执行和休眠，在有事情做的时候使当前NSRunLoop控制的线程工作，没有事情做让当前NSRunLoop的控制的线程休眠。
2.NSRunLoop 就是一直在循环检测，从线程start到线程end，检测inputsource(如点击，双击等操作)异步事件，检测timesource同步事件，检测到输入源会执行处理函数，首先会产生通知，corefunction向线程添加runloop observers来监听事件，意在监听事件发生时来做处理。
3.runloopmode是一个集合，包括监听：事件源，定时器，以及需通知的runloop observers。
4.只有在为你的程序创建次线程的时候，才需要运行run loop。对于程序的主线程而言，run loop是关键部分。Cocoa提供了运行主线程run loop的代码同时也会自动运行run loop。IOS程序UIApplication中的run方法在程序正常启动的时候就会启动run loop。如果你使用xcode提供的模板创建的程序，那你永远不需要自己去启动run loop。
5. 在多线程中，你需要判断是否需要run loop。如果需要run loop，那么你要负责配置run loop并启动。你不需要在任何情况下都去启动run loop。比如，你使用线程去处理一个预先定义好的耗时极长的任务时，你就可以毋需启动run loop。Run loop只在你要和线程有交互时才需要。

130. IOS7之前,后台执行内容有几种形式,都是什么

参考答案：一般的应用在进入后台的时候可以获取一定时间来运行相关任务，也就是说可以在后台运行一小段时间(10S左右)。

1. 后台播放音乐
2. 后台GPS跟踪
3. 后台voip支持

131. 简单说一下APP的启动过程,从main文件开始说起

参考答案：有storyboard情况下:

1.main函数

2.UIApplicationMain

a、创建UIApplication对象； b、创建UIApplication的delegate对象

3.根据Info.plist获得最主要storyboard的文件名,加载最主要的storyboard(有storyboard)

a、创建UIWindow； b、创建和设置UIWindow的rootViewController； c、显示窗口

没有storyboard情况下:

1.main函数

2.UIApplicationMain

a、创建UIApplication对象; b、创建UIApplication的delegate对象

3.delegate对象开始处理(监听)系统事件(没有storyboard)

a、程序启动完毕的时候, 就会调用代理的application:didFinishLaunchingWithOptions:方法

b、在application:didFinishLaunchingWithOptions:中创建UIWindow

c、创建和设置UIWindow的rootViewController;

d、显示窗口

132. 把程序自己关掉和程序进入后台,远程推送的区别

参考答案：1. 关掉后不执行任何代码, 不能处理事件

2. 应用程序进入后台状态不久后转入挂起状态。在这种状态下，应用程序不执行任何代码，并有可能在任意时候从内存中删除。只有当用户再次运行此应用，应用才会从挂起状态唤醒，代码得以继续执行

3.或者进入后台时开启多任务状态，保留在内存中，这样就可以执行系统允许的动作

4.远程推送是由远程服务器上的程序发送到APNS,再由APNS把消息推送至设备上的程序,当应用程序收到推送的消息会自动调用特定的方法执行事先写好的代码

133. 本地通知和远程推送通知对基本概念和用法?

参考答案：本地通知和远程推送通知都可以向不在前台运行的应用发送消息,这种消息既可能是即将发生的事件,也可能是服务器的新数据.不管是本地通知还是远程通知,他们在程序界面的显示效果相同,都可能显示为一段警告信息或应用程序图标上的微章.

本地通知和远程推送通知的基本目的都是让应用程序能够通知用户某些事情, 而且不需要应用程序在前台运行.二者的区别在于本地通知由本应用负责调用,只能从当前设备上的iOS发出, 而远程通知由远程服务器上的程序发送到APNS,再由APNS把消息推送至设备上的程序

134. 列举现在熟悉iOS开发库和第三方开发库?

参考答案：友盟（包括第三方登录和分享），高德地图，百度地图,AFN,SDWebimage,FMDB, MBProgressHUD等等

135. 有些图片加载的比较慢怎么处理?你是怎么优化程序的性能的?
参考答案：1> 图片下载放在异步线程

2> 图片下载过程中使用占位图片

3> 如果图片较大，可以考虑多线程断点下载

136. 你实现过一个框架或者库以供别人使用么？如果有，请谈一谈构建经验；如果没有，请设想和设计框架的API，并指出大概需要如何做、需要注意一些什么方面?
参考答案：1> 提供给外界的接口功能是否实用、够用

2> 别人使用我的框架时，能不能根据类名、方法名就猜出接口的具体作用

3> 别人调用接口时，提供的参数是否够用、调用起来是否简单

4> 别人使用我的框架时，要不要再导入依赖其他的框架

137. App需要加载超大量的数据，给服务器发送请求，但是服务器卡住了如何解决?
参考答案：1> 设置请求超时

2> 给用户提示请求超时

3> 根据用户操作再次请求数据

138. 你在实际开发中，有哪些手机架构与性能调试经验?
参考答案：刚接手公司的旧项目时，模块特别多，而且几乎所有的代码都写在控制器里面，比如UI控件代码、网络请求代码、数据存储代码

接下来采取MVC模式进行封装、重构

自定义UI控件封装内部的业务逻辑

封装网络请求工具类

封装数据存储工具类

139. 即时通讯中的大数据处理?
参考答案：用put上传到文件服务器，然后发带url的自定义格式的给对方，对方接收到之后下载

140. 如果在网络数据处理过程中,发现一处比较卡,一般怎么解决

参考答案：1> 检查网络请求操作是否被放在主线程了

2> 看看异步请求的数量是否太多了（子线程数量）

3> 数据量是否太大？如果太大，先清除一些不必要的对象（看不见的数据、图片）

4> 手机CPU使用率和内存问题

141. 怎么解决sqlite锁定的问题?

参考答案：1> 设置数据库锁定的处理函数
int sqlite3_busy_handler(sqlite3*, int(*)(void*,int), void*);

2> 设定锁定时的等待时间
int sqlite3_busy_timeout(sqlite3*, 60);

142. runtime怎么添加属性、方法等

参考答案：给系统的类添加属性的时候,可以使用runtime动态添加属性、方法。本质:动态添加属性,就是让某个属性与对象产生关联。runtime一般都是针对系统的类。
class_addIvar
class_addMethod
class_addProperty
class_addProtocol
class_replaceProperty

143. runtime如何实现weak属性?

参考答案：weak策略表明该属性定义了一种非拥有关系(nonowning relationship)。 为这种属性设置新值时，设置方法既不保留新值，也不释放旧值。此特质同assign类似;然而在属性所指的对象遭到摧毁时，属性值也会清空(nil out)。

runtime对注册的类，会进行布局，会将 weak 对象放入一个 hash 表中。用weak指向的对象内存地址作为key，当此对象的引用计数为0的时候会调用对象的 dealloc 方法，假设 weak 指向的对象内存地址是a，那么就会以a为key，在这个 weak hash表中搜索，找到所有以a为key的 weak 对象，从而设置为 nil。

在ARC环境无论是强指针还是弱指针都无需在 dealloc 设置为 nil， ARC 会自动帮我们处理即便是编译器不帮我们做这些，weak也不需要在dealloc中置nil。模拟下weak的setter方法，大致如下

```
- (void)setObject:(NSObject *)object{
    objc_setAssociatedObject(self, "object", object, OBJC_ASSOCIATION_ASSIGN);
    [object cyl_runAtDealloc:^{ _object = nil; }];
}
```

144. runtime如何通过selector找到对应的IMP地址？（分别考虑类方法和实例方法）

参考答案：每一个类对象中都一个对象方法列表（对象方法缓存）。
类方法列表是存放在类对象中isa指针指向的元类对象中（类方法缓存）。
方法列表中每个方法结构体中记录着方法的名称,方法实现,以及参数类型，其实selector本质就是方法名称,通过这个方法名称就可以在方法列表中找到对应的方法实现。
当我们发送一个消息给一个NSObject对象时，这条消息会在对象的类对象方法列表里查找。
当我们发送一个消息给一个类时，这条消息会在类的Meta Class对象的方法列表里查找。

145. 使用runtime Associate方法关联的对象，需要在主对象dealloc的时候释放么?

参考答案：无论在MRC下还是ARC下均不需要。被关联的对象在生命周期内要比对象本身释放的晚很多，它们会在被 NSObject -dealloc 调用的object_dispose()方法中释放。

146. 对象的内存销毁时间表，分哪几个步骤?

参考答案：1、调用 -release ： 引用计数变为零；对象正在被销毁，生命周期即将结束；不能再有新的 __weak 弱引用，否则将指向 nil；调用 [self dealloc]

2、 父类调用 -dealloc：继承关系中最直接继承的父类再调用 -dealloc；如果是 MRC 代码 则会手动释放实例变量们（iVars；继承关系中每一层的父类 都再调用 -dealloc

3、NSObject 调 -dealloc ： 调用 Objective-C runtime 中object_dispose() 方法

4. 调用 object_dispose()：为 C++ 的实例变量们（iVars）调用 destructors；为 ARC 状态下的 实例变量们（iVars）调用 -release；解除所有使用 runtime Associate方法关联的对象；解除所有 __weak 引用；调用 free()

147. _objc_msgForward函数是做什么的？直接调用它将会发生什么?

参考答案：_objc_msgForward是IMP类型，用于消息转发的： 当向一个对象发送一条消息，但它并没有实现的时候，_objc_msgForward会尝试做消息转发。 直接调用_objc_msgForward是非常危险的事，这是把双刃刀，如果用不好会直接导致程序Crash，但是如果用得好，能做很多非常酷的事。

148. 能否向编译后得到的类中增加实例变量？能否向运行时创建的类中添加实例变量？为什么?

参考答案：不能向编译后得到的类中增加实例变量；
能向运行时创建的类中添加实例变量；

分析如下：
因为编译后的类已经注册在runtime中，类结构体中的objc_ivar_list 实例变量的链表和instance_size实例变量的内存大小已经确定，同时runtime 会调用class_setIvarLayout 或 class_setWeakIvarLayout来处理strong weak引用，所以不能向存在的类中添加实例变量运行时创建的类是可以添加实例变量，调用 class_addIvar函数，但是得在调用objc_allocateClassPair之后，objc_registerClassPair之前，原因同上。

149. 简述下Objective-C中调用方法的过程（runtime）。

参考答案：Objective-C是动态语言，每个方法在运行时会被动态转为消息发送，即：objc_msgSend(receiver, selector)，整个过程介绍如下：
objc在向一个对象发送消息时，runtime库会根据对象的isa指针找到该对象实际所属的类。
然后在该类中的方法列表以及其父类方法列表中寻找方法运行。
如果在最顶层的父类（一般也就NSObject）中依然找不到相应的方法时，程序在运行时会挂掉并抛出异常unrecognized selector sent to XXX。
但是在这之前，objc的运行时会给出三次拯救程序崩溃的机会。
补充说明：Runtime 铸就了Objective-C 是动态语言的特性，使得C语言具备了面向对象的特性，在程序运行期创建，检查，修改类、对象及其对应的方法，这些操作都可以使用runtime中的对应方法实现。

150. 什么是method swizzling?

参考答案：简单说就是进行方法交换
在Objective-C中调用一个方法， 其实是向一个对象发送消息，查找消息的唯一依据是selector的名字。利用Objective-C的动态特性，可以实现在运行时偷换selector对应的方法实现，达到给方法挂钩的目的。
每个类都有一个方法列表，存放着方法的名字和方法实现的映射关系，selector的本质其实就是方法名，IMP有点类似函数指针，指向具体的Method实现，通过selector就可以找到对应的IMP。

交换方法的几种实现方式：
利用 method_exchangeImplementations 交换两个方法的实现
利用 class_replaceMethod 替换方法的实现
利用 method_setImplementation 来直接设置某个方法的IMP

151. Objective-C类方法load和initialize的区别?

参考答案：initialize和load的区别在于：

load是只要类所在文件被引用就会被调用，而initialize是在类或者其子类的第一个方法被调用前调用。所以如果类没有被引用进项目，就不会有load调用；但即使类文件被引用进来，但是没有使用，那么initialize也不会被调用。

它们的相同点在于：方法只会被调用一次。

方法调用的顺序：父类(Superclass)的方法优先于子类(Subclass)的方法，类中的方法优先于类别(Category)中的方法。

152. 对数组中的元素去重复：NSArray *array = @[@"12-11", @"12-11", @"12-11", @"12-12", @"12-13", @"12-14"];

参考答案：

```
NSMutableArray *resultArray = [[NSMutableArray alloc]    initWithCapacity:array.count];
// 外层一个循环
for (NSString *item in array) {
        // 调用-containsObject:本质也是要循环去判断，因此本质上是双层遍历
        // 时间复杂度为O ( n^2 )而不是O (n)
        if (![resultArray containsObject:item]) {
                [resultArray addObject:item];
        }
}
NSLog(@"resultArray: %@", resultArray);
```

153. 使用集合对数组中的元素去重复：NSArray *array = @[@"12-11", @"12-11", @"12-11", @"12-12", @"12-13", @"12-14"];

参考答案：集合操作可以通过valueForKeyPath来实现的，去重可以一行代码实现：

```
array = [array valueForKeyPath:@"@distinctUnionOfObjects.self"];
NSLog(@"%@", array);
```

但是返回的结果是无序的，与原来的顺序不同。

154. 使用字典对数组中的元素去重复：NSArray *array = @[@"12-11", @"12-11", @"12-11", @"12-12", @"12-13", @"12-14"];

参考答案：利用NSDictionary去重，字典在设置key-value时，若已存在则更新值，若不存在则插入值，然后获取allValues。

若不要求有序，则可以采用此种方法。若要求有序，还得进行排序。

效率分析：只需要一个循环就可以完成放入字典，若不要求有序，时间复杂度为O(n)。

若要求排序，则效率与排序算法有关：

```
NSMutableDictionary *resultDict = [[NSMutableDictionary alloc] initWithCapacity:array.count];
for (NSString *item in array) {
        [resultDict setObject:item forKey:item];
}
NSArray *resultArray = resultDict.allValues;
NSLog(@"%@", resultArray);
```

155. 使用NSSet对数组中的元素去重复：NSArray *array = @[@"12-11", @"12-11", @"12-11", @"12-12", @"12-13", @"12-14"];
参考答案：利用集合NSSet的特性(确定性、无序性、互异性)，放入集合就自动去重了。但是它与字典拥有同样的无序性，所得结果顺序不再与原来一样。

```
NSSet *set = [NSSet setWithArray:array];
NSArray *resultArray = [set allObjects];
NSLog(@"%@", resultArray);
```

如果要求有序，那就得排序，比如这里要升序排序：

```
resultArray = [resultArray sortedArrayUsingComparator:^NSComparisonResult(id _Nonnull obj1, id _Nonnull obj2) {
        NSString *item1 = obj1;
        NSString *item2 = obj2;
        return [item1 compare:item2 options:NSLiteralSearch];
}];
NSLog(@"%@", resultArray);
```

156. 使用NSOrderedSet对数组中的元素去重复：NSArray *array = @[@"12-11", @"12-11", @"12-11", @"12-12", @"12-13", @"12-14"];
参考答案：NSOrderedSet *set = [NSOrderedSet orderedSetWithArray:array];

```
NSLog(@"%@", set.array);
```

157. 谈谈你对NSArray和NSMutableArray的理解。
参考答案：NSArray表示不可变数组，是有序元素集，只能存储对象类型，可通过索引直接访问元素，而且元素类型可以不一样，但是不能进行增、删、改操作；

NSMutableArray是可变数组，能进行增、删、改操作。通过索引查询值很快，但是插入、删除等效率很低。
数组用于处理一组有序的数据集，比如常用的列表的dataSource要求有序，可通过索引直接访问，效率高。

158. 谈谈你对NSSet和NSMutableSet的理解。
参考答案：NSSet表示不可变集合，具有确定性、互异性、无序性的特点，只能访问而不能修改集合；
NSMutableSet表示可变集合，可以对集合进行增、删、改操作。集合通过值查询很快，插入、删除操作极快。

159. 谈谈你对NSDictionary和NSMutableDictionary的理解。
参考答案：NSDictionary表示不可变字典，具有无序性的特点，每个key对应的值是唯一的，可通过key直接获取值；
NSMutableDictionary表示可变字典，能对字典进行增、删、改操作。通过key查询值、插入、删除值都很快。

160. 简单描述一下XIB与Storyboards，说一下他们的优缺点。
参考答案：优点： XIB：在编译前就提供了可视化界面，可以直接拖控件，也可以直接给控件添加约束，更直观一些，而且类文件中就少了创建控件的代码，确实简化不少，通常每个XIB对应一个类。
Storyboard：在编译前提供了可视化界面，可拖控件，可加约束，在开发时比较直观，而且一个storyboard可以有很多的界面，每个界面对应一个类文件，通过storybard，可以直观地看出整个App的结构。

缺点：
XIB：需求变动时，需要修改XIB很大，有时候甚至需要重新添加约束，导致开发周期变长。XIB载入相比纯代码自然要慢一些。对于比较复杂逻辑控制不同状态下显示不同内容时，使用XIB是比较困难的。当多人团队或者多团队开发时，如果XIB文件被发动，极易导致冲突，而且解决冲突相对要困难很多。
Storyboard：需求变动时，需要修改storyboard上对应的界面的约束，与XIB一样可能要重新添加约束，或者添加约束会造成大量的冲突，尤其是多团队开发。对于复杂逻辑控制不同显示内容时，比较困难。当多人团队或者多团队开发时，大家会同时修改一个storyboard，导致大量冲突，解决起来相当困难。

面试题第三篇

共有80题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

161. 请把字符串2018-03-11格式化日期转为NSDate类型

参考答案：NSString *timeStr = @"2018-03-11";
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
formatter.dateFormat = @"yyyy-MM-dd";
formatter.timeZone = [NSTimeZone defaultTimeZone];
NSDate *date = [formatter dateFromString:timeStr];
NSLog(@"%@", date);

162. 在App中混合HTML5开发App如何实现的。在App中使用HTML5的优缺点是什么?

参考答案：在iOS中，通常是通常UIWebView来实现，当然在iOS8以后可以使用WKWebView来实现.有以下几种实现方法：
通过实现UIWebView的代理方法来拦截，判断scheme是否是约定好的，然后iOS调用本地相关API来实现：

- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request navigationType:(UIWebViewNavigationType)navigationType; 在iOS7以后，可以直接通过JavaScripteCore这个库来实现，通过往JS DOM注入对象，而这个对象对应于我们iOS的某个类的实例。iOS加入H5响应比原生要慢很多，体验不太好，这是缺点。

iOS加入H5可以实现嵌入别的功能入口，可随时更改，不用更新版本就可以上线，这是最大的优点。

163. 请描述一下同步和异步，说说它们之间的区别。

参考答案：首先，我们要明确一点，同步和异步都是在线程中使用的。 在iOS开发中，比如网络请求数据时，若使用同步请求，则只有请求成功或者请求失败得到响应返回后，才能继续往下走，也就是才能访问其它资源（会阻塞了线程）。
网络请求数据异步请求时，不会阻塞线程，在调用请求后，可以继续往下执行，而不用等请求有结果才能继续。

区别：
线程同步：是多个线程访问同一资源时，只有当前正在访问的线程访问结束之后，其他线程才能开始访问（被阻塞）。
线程异步：是多个线程在访问竞争资源时，可以在空闲等待时去访问其它资源（不被阻塞）。

164. 述一下iOS的内存管理，在开发中对于内存的使用和优化包含哪些方面。我们在开发中应该注意哪些问题。

参考答案：对于内存的使用和优化常见的有以下方面：
重用问题：如UITableViewCells、UICollectionViewCells、UITableViewHeaderFooterViews设置正确的reuseIdentifier，充分重用。
尽量把views设置为不透明：当opque为NO的时候，图层的半透明取决于图片和其本身合成的图层为结果，可提高性能。
不要使用太复杂的XIB/Storyboard：载入时就会将XIB/storyboard需要的所有资源，包括图片全部载入内存，即使未来很久才会使用。那些相比纯代码写的延迟加载，性能及内存就差了很多。
选择正确的数据结构：学会选择对业务场景最合适的数组结构是写出高效代码的基础。比如，数组: 有序的一组值。使用索引来查询很快，使用值查询很慢，插入/删除很慢。字典: 存储键值对，用键来查找比较快。集合: 无序的一组值，用值来查找很快，插入/删除很快。
gzip/zip压缩：当从服务端下载相关附件时，可以通过gzip/zip压缩后再下载，使得内存更小，下载速度也更快。
延迟加载：对于不应该使用的数据，使用延迟加载方式。对于不需要马上显示的视图，使用延迟加载方式。比如，网络请求失败时显示的提示界面，可能一直都不会使用到，因此应该使用延迟加载。
数据缓存：对于cell的行高要缓存起来，使得reload数据时，效率也极高。而对于那些网络数据，不需要每次都请求的，应该缓存起来，可以写入数据库，也可以通过plist文件存储。
重用大开销对象：一些objects的初始化很慢，比如NSDateFormatter和NSCalendar，但又不可避免地需要使用它们。通常是作为属性存储起来，防止反复创建。
避免反复处理数据：许多应用需要从服务器加载功能所需的常为JSON或者XML格式的数据。在服务器端和客户端使用相同的数据结构很重要。
使用Autorelease Pool：在某些循环创建临时变量处理数据时，自动释放池以保证能及时释放内存。

165. plist文件是用来做什么的。一般用它来处理一些什么方面的问题。
参考答案：plist是iOS系统中特有的文件格式。我们常用的NSUserDefaults偏好设置实质上就是plist文件操作。plist文件是用来持久化存储数据的。

我们通常使用它来存储偏好设置，以及那些少量的、数组结构比较复杂的不适合存储数据库的数据。比如，我们要存储全国城市名称和id，那么我们要优先选择plist直接持久化存储，因为更简单。

166. iOS中缓存一定量的数据以便下次可以快速执行，那么数据会存储在什么地方，有多少种存储方式?
参考答案：偏好设置(NSUserDefaults)
plist文件存储
归档
SQLite3
Core Data

167. 在提交苹果审核时，遇到哪些问题被拒绝，对于被拒绝的问题是如何处理的?
参考答案：app中有虚拟物品交易，但是没走内购导致被拒。
音频类App或者使用到音频相关的app，因为版权问题而被拒
App出现必闪退而被拒

168. Objective-C Runtime 是什么?
参考答案：Objective-C 的 Runtime 是一个运行时库（Runtime Library），它是一个主要使用 C 和汇编写的库，为 C 添加了面相对象的能力并创造了 Objective-C。
这就是说它在类信息（Class information） 中被加载，完成所有的方法分发，方法转发，等等。
Objective-C runtime 创建了所有需要的结构体，让 Objective-C 的面相对象编程变为可能。

169. Method Swizzling 原理
参考答案：在Objective-C中调用一个方法，其实是向一个对象发送消息，查找消息的唯一依据是selector的名字。利用Objective-C的动态特性，可以实现在运行时偷换selector对应的方法实现，达到给方法挂钩的目的。每个类都有一个方法列表，存放着selector的名字和方法实现的映射关系。IMP有点类似函数指针，指向具体的Method实现。

我们可以利用 method_exchangeImplementations 来交换2个方法中的IMP,

我们可以利用 class_replaceMethod 来修改类,

我们可以利用 method_setImplementation 来直接设置某个方法的IMP,

归根结底，都是偷换了selector的IMP。

170. 深浅复制和属性为copy，strong值的变化问题

参考答案：浅复制：只复制指向对象的指针，而不复制引用对象本身。对于浅复制来说，A和A_copy指向的是同一个内存资源，复制的只不个是一个指针，对象本身资源还是只有一份，那如果我们对A_copy执行了修改操作，那么发现A引用的对象同样被修改了。深复制就好理解了，内存中存在了两份独立对象本身。

[immutableObject copy] // 浅拷贝
[immutableObject mutableCopy] //深拷贝
[mutableObject copy] //深拷贝
[mutableObject mutableCopy] //深拷贝
属性设为copy,指定此属性的值不可更改，防止可变字符串更改自身的值的时候不会影响到对象属性（如NSString,NSArray,NSDictionary）的值。strong此属性的指会随着变化而变化。copy是内容拷贝，strong是指针拷贝。

171. NSTimer创建后，会在哪个线程运行?

参考答案：用scheduledTimerWithTimeInterval创建的，在哪个线程创建就会被加入哪个线程的RunLoop中就运行在哪个线程。

自己创建的Timer，加入到哪个线程的RunLoop中就运行在哪个线程。

172. KVO，NSNotification，delegate及block区别是什么?

参考答案：KVO就是cocoa框架实现的观察者模式，一般同KVC搭配使用，通过KVO可以监测一个值的变化，比如View的高度变化。是一对多的关系，一个值的变化会通知所有的观察者。

NSNotification是通知，也是一对多的使用场景。在某些情况下，KVO和NSNotification是一样的，都是状态变化之后告知对方。NSNotification的特点，就是需要被观察者先主动发出通知，然后观察者注册监听后再来进行响应，比KVO多了发送通知的一步，但是其优点是监听不局限于属性的变化，还可以对多种多样的状态变化进行监听，监听范围广，使用也更灵活。

delegate 是代理，就是我不想做的事情交给别人做。比如狗需要吃饭，就通过delegate通知主人，主人就会给他做饭、盛饭、倒水，这些操作，这些狗都不需要关心，只需要调用delegate（代理人）就可以了，由其他类完成所需要的操作。所以delegate是一对一关系。

block是delegate的另一种形式，是函数式编程的一种形式。使用场景跟delegate一样，相比delegate更灵活，而且代理的实现更直观。

delegate一般的使用场景是行为，需求是需要别人帮我做一件事情，比如买卖股票，我们一般使用delegate。Notification一般是进行全局通知。delegate是强关联，就是委托和代理双方互相知道。Notification是弱关联，利好消息发出，你不需要知道是谁发的也可以做出相应的反应，同理发消息的人也不需要知道接收的人也可以正常发出消息。

173. 如何让计时器调用一个类方法

参考答案：计时器只能调用实例方法，但是可以在这个实例方法里面调用静态方法。

使用计时器需要注意，计时器一定要加入RunLoop中，并且选好model才能运行。scheduledTimerWithTimeInterval方法创建一个计时器并加入到RunLoop中所以可以直接使用。

如果计时器的repeats选择YES说明这个计时器会重复执行，一定要在合适的时机调用计时器的invalid。不能在dealloc中调用，因为一旦设置为repeats 为yes，计时器会强持有self，导致dealloc永远不会被调用，这个类就永远无法被释放。

比如可以在viewDidDisappear中调用，这样当类需要被回收的时候就可以正常进入dealloc中了。

174. 调用一个类的静态方法需不需要release?

参考答案：静态方法，就是类方法，不需要，类方法对象放在autorelease中。

175. static作用?

参考答案： （1）函数体内 static 变量的作用范围为该函数体，不同于 auto 变量，该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值；

（2）在模块内的 static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；

（3）在模块内的 static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明它的模块内；

（4）在类中的 static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；

（5）在类中的 static 成员函数属于整个类所拥有，这个函数不接收 this 指针，因而只能访问类的static 成员变量。

176. 使用递归方式，反转二叉树。

参考答案： public class TreeNode {

```
    int val;TreeNode left;TreeNode right;TreeNode(int x) { val = x; }
}
public class Solution {
    public TreeNode invertTree(TreeNode root) {
        if (root == null) {
                return null;
        }
        root.left = invertTree(root.left);
        root.right = invertTree(root.right);
        TreeNode tmp = root.left;
        root.left = root.right; root.right = tmp;
        return root;
    }
}
```

177. 不要使用递归方式，反转二叉树。

参考答案： + (BinaryTreeNode *)invertBinaryTree:(BinaryTreeNode *)rootNode {

```
        if (!rootNode) { return nil; }
        if (!rootNode.leftNode && !rootNode.rightNode) { return rootNode; }
        NSMutableArray *queueArray = [NSMutableArray array]; //数组当成队列
        [queueArray addObject:rootNode]; //压入根节点
        while (queueArray.count > 0) {
                BinaryTreeNode *node = [queueArray firstObject];
                [queueArray removeObjectAtIndex:0]; //弹出最前面的节点，仿照队列先进先出原则
                BinaryTreeNode *pLeft = node.leftNode;
                node.leftNode = node.rightNode;
                node.rightNode = pLeft;
                if (node.leftNode) {[queueArray addObject:node.leftNode];}
                if (node.rightNode) {[queueArray addObject:node.rightNode];}
        }
    return rootNode;
}
```

178. 使用Objective-C编写一个单例

参考答案：

```
+ (AccountManager *)sharedManager
{
    static AccountManager *sharedAccountManagerInstance = nil;
    tatic dispatch_once_t predicate;
    dispatch_once(&predicate, ^{
        sharedAccountManagerInstance = [[self alloc] init];
    });
    return sharedAccountManagerInstance;
}
```

179. 使用Swift编写一个单例

参考答案：

```
final class SingleClass: NSObject
{
    static let shared = SingleClass()
    private override init() {}

    func say()
    {
        print("Hello, CoolKeTang!")
    }
}
```

180. iOS应用程序的状态有哪些?

参考答案：Not running未运行：程序没启动。

Inactive未激活：程序在前台运行，不过没有接收到事件。在没有事件处理情况下程序通常停留在这个状态。

Active激活：程序在前台运行而且接收到了事件。这也是前台的一个正常的模式。

Backgroud后台：程序在后台而且能执行代码，大多数程序进入这个状态后会在在这个状态上停留一会。时间到之后会进入挂起状态(Suspended)。有的程序经过特殊的请求后可以长期处于Backgroud状态。

Suspended挂起：程序在后台不能执行代码。系统会自动把程序变成这个状态而且不会发出通知。当挂起时，程序还是停留在内存中的，当系统内存低时，系统就把挂起的程序清除掉，为前台程序提供更多的内存。

181. 网络请求中post和get的区别

参考答案：1.GET使用URL或Cookie传参。而POST将数据放在BODY中。

2.GET的URL会有长度上的限制，则POST的数据则可以非常大。

3.POST比GET安全，因为数据在地址栏上不可见。

182. 什么是事件响应链Responder Chain?

参考答案：响应者对象（Responder Object），指的是有响应和处理事件能力的对象。响应者链就是由一系列的响应者对象构成的一个层次结构。

UIResponder是所有响应对象的基类，在UIResponder类中定义了处理上述各种事件的接口。我们熟悉的UIApplication、 UIViewController、UIWindow和所有继承自UIView的UIKit类都直接或间接的继承自UIResponder，所以它们的实例都是可以构成响应者链的响应者对象。

响应者链有以下特点：
1、响应者链通常是由视图（UIView）构成的；
2、一个视图的下一个响应者是它视图控制器（UIViewController）（如果有的话），然后再转给它的父视图（Super View）；
3、视图控制器（如果有的话）的下一个响应者为其管理的视图的父视图；
4、单例的窗口（UIWindow）的内容视图将指向窗口本身作为它的下一个响应者
5、单例的应用（UIApplication）是一个响应者链的终点，它的下一个响应者指向nil，以结束整个循环。

183. 点击屏幕时是如何互动的

参考答案：iOS系统检测到手指触摸(Touch)操作时会将其打包成一个UIEvent对象，并放入当前活动Application的事件队列，单例的UIApplication会从事件队列中取出触摸事件并传递给单例的UIWindow来处理，UIWindow对象首先会使用hitTest:withEvent:方法寻找此次Touch操作初始点所在的视图(View)，即需要将触摸事件传递给其处理的视图，这个过程称之为hit-test view。

UIWindow实例对象会首先在它的内容视图上调用hitTest:withEvent:，此方法会在其视图层级结构中的每个视图上调用pointInside:withEvent:（该方法用来判断点击事件发生的位置是否处于当前视图范围内，以确定用户是不是点击了当前视图），如果pointInside:withEvent:返回YES，则继续逐级调用，直到找到touch操作发生的位置，这个视图也就是要找的hit-test view。

事件的传递和响应分两个链：
传递链：由系统向离用户最近的view传递。UIKit > active app’s event queue > window > root view >lowest view
响应链：由离用户最近的view向系统传递。initial view > super view > view controller > window > Application

184. Run Loop是什么，使用的目的，何时使用和关注点

参考答案：Run Loop是一让线程能随时处理事件但不退出的机制。RunLoop 实际上是一个对象，这个对象管理了其需要处理的事件和消息，并提供了一个入口函数来执行Event Loop 的逻辑。线程执行了这个函数后，就会一直处于这个函数内部 “接受消息->等待->处理” 的循环中，直到这个循环结束（比如传入 quit 的消息），函数返回。让线程在没有处理消息时休眠以避免资源占用、在有消息到来时立刻被唤醒。

OSX/iOS 系统中，提供了两个这样的对象：NSRunLoop 和 CFRunLoopRef。CFRunLoopRef 是在 CoreFoundation 框架内的，它提供了纯 C 函数的 API，所有这些 API 都是线程安全的。NSRunLoop 是基于 CFRunLoopRef 的封装，提供了面向对象的 API，但是这些 API 不是线程安全的。

线程和 RunLoop 之间是一一对应的，其关系是保存在一个全局的 Dictionary 里。线程刚创建时并没有 RunLoop，如果你不主动获取，那它一直都不会有。RunLoop 的创建是发生在第一次获取时，RunLoop 的销毁是发生在线程结束时。你只能在一个线程的内部获取其 RunLoop（主线程除外）。

185. Run Loop有哪些作用?

参考答案：使程序一直运行接受用户输入
决定程序在何时应该处理哪些Event
调用解耦
节省CPU时间

186. 什么时候使用run loop?

参考答案：仅当在为你的程序创建辅助线程的时候，你才需要显式运行一个run loop。Run loop是程序主线程基础设施的关键部分。所以，Cocoa和Carbon程序提供了代码运行主程序的循环并自动启动run loop。IOS程序中UIApplication的run方法（或Mac OS X中的NSApplication）作为程序启动步骤的一部分，它在程序正常启动的时候就会启动程序的主循环。类似的，RunApplicationEventLoop函数为Carbon程序启动主循环。如果你使用xcode提供的模板创建你的程序，那你永远不需要自己去显式的调用这些例程。

对于辅助线程，你需要判断一个run loop是否是必须的。如果是必须的，那么你要自己配置并启动它。你不需要在任何情况下都去启动一个线程的run loop。比如，你使用线程来处理一个预先定义的长时间运行的任务时，你应该避免启动run loop。Run loop在你要和线程有更多的交互时才需要，比如以下情况：

使用端口或自定义输入源来和其他线程通信
使用线程的定时器
Cocoa中使用任何performSelector…的方法
使线程周期性工作

187. 常用的多线程处理方式及优缺点是什么?

参考答案：iOS有四种多线程编程的技术，分别是：NSThread，Cocoa NSOperation，GCD和pthread。

1）NSThread优点：NSThread 比其他两个轻量级。缺点：需要自己管理线程的生命周期，线程同步。线程同步对数据的加锁会有一定的系统开销。
2）Cocoa NSOperation优点:不需要关心线程管理， 数据同步的事情，可以把精力放在自己需要执行的操作上。Cocoa operation相关的类是NSOperation, NSOperationQueue.NSOperation是个抽象类,使用它必须用它的子类，可以实现它或者使用它定义好的两个子类: NSInvocationOperation和NSBlockOperation.创建NSOperation子类的对象，把对象添加到NSOperationQueue队列里执行。
3)GCD(全优点)Grand Central dispatch(GCD)是Apple开发的一个多核编程的解决方案。在iOS4.0开始之后才能使用。GCD是一个替代NSThread, NSOperationQueue,NSInvocationOperation等技术的很高效强大的技术。
4) pthread是一套通用的多线程API，适用于Linux\Windows\Unix,跨平台，可移植，使用C语言，生命周期需要程序员管理，IOS开发中使用很少。

188. dispatch_sync 和 dispatch_async 区别是什么?

参考答案：dispatch_async(queue,block) async 异步队列，dispatch_async函数会立即返回, block会在后台异步执行。

dispatch_sync(queue,block) sync 同步队列，dispatch_sync函数不会立即返回，及阻塞当前线程,等待 block同步执行完成。

189. 请说明并比较以下关键词：__weak，__block

参考答案：__weak与weak基本相同。前者用于修饰变量（variable），后者用于修饰属性（property）。__weak主要用于防止block中的循环引用。

__block也用于修饰变量。它是引用修饰，所以其修饰的值是动态变化的，即可以被重新赋值的。__block用于修饰某些block内部将要修改的外部变量。

__weak和__block的使用场景几乎与block息息相关。而所谓block，就是Objective-C对于闭包的实现。闭包就是没有名字的函数，或者理解为指向函数的指针。

190. 什么是ARC?

参考答案：ARC全称是 Automatic Reference Counting，是Objective-C的内存管理机制。简单地来说，就是代码中自动加入了retain/release，原先需要手动添加的用来处理内存管理的引用计数的代码可以自动地由编译器完成了。

ARC的使用是为了解决对象retain和release匹配的问题。以前手动管理造成内存泄漏或者重复释放的问题将不复存在。

以前需要手动的通过retain去为对象获取内存，并用release释放内存。所以以前的操作称为MRC (Manual Reference Counting)。

191. 以scheduledTimerWithTimeInterval的方式触发的timer，在滑动页面上的列表时，timer会暂停，为什么？该如何解决?

参考答案：原因在于滑动时当前线程的runloop切换了mode用于列表滑动，导致timer暂停。
runloop中的mode主要用来指定事件在runloop中的优先级，有以下几种：
Default（NSDefaultRunLoopMode）：默认，一般情况下使用；
Connection（NSConnectionReplyMode）：一般系统用来处理NSConnection相关事件，开发者一般用不到；
Modal（NSModalPanelRunLoopMode）：处理modal panels事件；
Event Tracking（NSEventTrackingRunLoopMode）：用于处理拖拽和用户交互的模式。
Common（NSRunloopCommonModes）：模式合集。默认包括Default，Modal，Event Tracking三大模式，可以处理几乎所有事件。
回到题中的情境。滑动列表时，runloop的mode由原来的Default模式切换到了Event Tracking模式，timer原来好好的运行在Default模式中，被关闭后自然就停止工作了。
解决方法其一是将timer加入到NSRunloopCommonModes中。其二是将timer放到另一个线程中，然后开启另一个线程的runloop，这样可以保证与主线程互不干扰，而现在主线程正在处理页面滑动。

192. Swift的类（class）和结构体（struct）有什么区别?

参考答案：Swift 中，类是引用类型，结构体是值类型。

值类型在传递和赋值时将进行复制，而引用类型则只会使用引用对象的一个"指向"。

所以他们两者之间的区别就是两个类型的区别。

193. Swift的class有哪些功能是struct没有的?

参考答案：class可以继承，这样子类可以使用父类的特性和方法

类型转换可以在runtime的时候检查和解释一个实例的类型

可以用deinit来释放资源

一个类可以被多次引用

194. Swift的struct相对于class有什么优势?

参考答案：结构较小，适用于复制操作，相比于一个class的实例被多次引用更加安全。

无须担心内存memory leak或者多线程冲突问题

195. Swift 是面向对象还是函数式的编程语言?

参考答案：Swift 既是面向对象的，又是函数式的编程语言。

说 Swift 是面向对象的语言，是因为 Swift 支持类的封装、继承、和多态，从这点上来看与 Java 这类纯面向对象的语言几乎毫无差别。

说 Swift 是函数式编程语言，是因为 Swift 支持 map, reduce, filter, flatmap 这类去除中间状态、数学函数式的方法，更加强调运算结果而不是中间过程。

196. 请说明并比较Swift语言中的一些关键词：Open, Public, Internal, File-private, Private

参考答案：Swift 有五个级别的访问控制权限，从高到底依次为比如 Open, Public, Internal, File-private, Private。

他们遵循的基本原则是：高级别的变量不允许被定义为低级别变量的成员变量。比如一个 private 的 class 中不能含有 public 的 String。反之，低级别的变量却可以定义在高级别的变量中。比如 public 的 class 中可以含有 private 的 Int。

Open 具备最高的访问权限。其修饰的类和方法可以在任意 Module 中被访问和重写；它是 Swift 3 中新添加的访问权限。

Public 的权限仅次于 Open。与 Open 唯一的区别在于它修饰的对象可以在任意 Module 中被访问，但不能重写。

Internal 是默认的权限。它表示只能在当前定义的 Module 中访问和重写，它可以被一个 Module 中的多个文件访问，但不可以被其他的 Module 中被访问。

File-private 也是 Swift 3 新添加的权限。其被修饰的对象只能在当前文件中被使用。例如它可以被一个文件中的 class，extension，struct 共同使用。

Private 是最低的访问权限。它的对象只能在定义的作用域内使用。离开了这个作用域，即使是同一个文件中的其他作用域，也无法访问。

197. 请说明并比较Swift语言中的以下关键词：strong, weak, unowned

参考答案：Swift 的内存管理机制与 Objective-C一样为 ARC（Automatic Reference Counting）。它的基本原理是，一个对象在没有任何强引用指向它时，其占用的内存会被回收。反之，只要有任何一个强引用指向该对象，它就会一直存在于内存中。

strong 代表着强引用，是默认属性。当一个对象被声明为 strong 时，就表示父层级对该对象有一个强引用的指向。此时该对象的引用计数会增加1。

weak 代表着弱引用。当对象被声明为 weak 时，父层级对此对象没有指向，该对象的引用计数不会增加1。它在对象释放后弱引用也随即消失。继续访问该对象，程序会得到 nil，不亏崩溃

unowned 与弱引用本质上一样。唯一不同的是，对象在释放后，依然有一个无效的引用指向对象，它不是 Optional 也不指向 nil。如果继续访问该对象，程序就会崩溃。

198. Swift语言中的weak 和 unowned 的使用场景有哪些差别?

参考答案：当访问对象时该对象可能已经被释放了，则用 weak。比如 delegate 的修饰。

当访问对象确定不可能被释放，则用 unowned。比如 self 的引用。

实际上为了安全起见，很多公司规定任何时候都使用 weak 去修饰。

199. 请用 Swift 实现或（||）操作

参考答案： func ||(left: Bool, right: Bool) –> Bool {
if left {
return true
} else {
return right
}
}

200. 使用Swift实现一个函数。求一个整型二维数组中所有元素之和?

参考答案： func sumPairs(_ nums: [[Int]]) -> Int {
return nums.flatMap { $0 }.reduce(0) { $0 + $1 }
}
Swift 有函数式编程的思想。其中 flatMap, map, reduce, filter 是其代表的方法。本题中考察了 flatMap 的降维思路，以及 reduce 的基本使用。相比于一般的 for 循环，这样的写法要更加得简洁漂亮。

201. 说说Swift为什么将String，Array，Dictionary设计成值类型?

参考答案：要解答这个问题，就要和Objective-C中相同的数据结构设计进行比较。Objective-C中，字符串，数组，字典，皆被设计为引用类型。

值类型相比引用类型，最大的优势在于内存使用的高效。值类型在栈上操作，引用类型在堆上操作。栈上的操作仅仅是单个指针的上下移动，而堆上的操作则牵涉到合并、移位、重新链接等。也就是说Swift这样设计，大幅减少了堆上的内存分配和回收的次数。同时copy-on-write又将值传递和复制的开销降到了最低。

String，Array，Dictionary设计成值类型，也是为了线程安全考虑。通过Swift的let设置，使得这些数据达到了真正意义上的“不变”，它也从根本上解决了多线程中内存访问和操作顺序的问题。

设计成值类型还可以提升API的灵活度。例如通过实现Collection这样的协议，我们可以遍历String，使得整个开发更加灵活高效。

202. 在Swift和Objective-C的混编项目中，如何在Swift文件中调用Objective-C文件中已经定义的方法？如何在Objective-C文件中调用Swift文件中定义的方法?

参考答案：Swift中若要使用Objective-C代码，可以在ProjectName-Bridging-Header.h里添加Objective-C的头文件名称，Swift文件中即可调用相应的Objective-C代码。一般情况Xcode会在Swift项目中第一次创建Objective-C文件时自动创建ProjectName-Bridging-Header.h文件。

Objective-C中若要调用Swift代码，可以导入Swift生成的头函数ProjectName-Swift.h来实现。

Swift文件中若要规定固定的方法或属性暴露给Objective-C使用，可以在方法或属性前加上@objc来声明。如果该类是NSObject子类，那么Swift会在非private的方法或属性前自动加上@objc。

203. 用Swift 将协议（protocol）中的部分方法设计成可选（optional），该怎样实现?

参考答案：@optional 和 @required 是 Objective-C 中特有的关键字。

Swift中，默认所有方法在协议中都是必须实现的。而且，协议里方法不可以直接定义 optional。先给出两种解决方案：

在协议和方法前都加上 @objc 关键字，然后再在方法前加上 optional 关键字。该方法实际上是把协议转化为Objective-C的方式然后进行可选定义。第2种方式是用扩展（extension）来规定可选方法。Swift中，协议扩展（protocol extension）可以定义部分方法的默认实现，这样这些方法在实际调用中就是可选实现的了。

204. 要给一个UIButton增加一个点击后抖动的效果，该怎样实现?
参考答案：实现一个自定义的UIButton类，在其中添加点击抖动效果的方法（shake方法）

写一个UIButton或者UIView的拓展（extension），然后在其中增加shake方法

定义一个protocol，然后在协议扩展（protocol extension）中添加shake方法

205. 试比较Swift和Objective-C中的初始化方法（init）有什么异同?
参考答案：Swift中的初始化方法更加严格和准确。

Objective-C中，初始化方法无法保证所有成员变量都完成初始化；编译器对属性设置并无警告，但是实际操作中会出现初始化不完全的问题；初始化方法与普通方法并无实际差别，可以多次调用。

Swift中，初始化方法必须保证所有optional的成员变量都完成初始化。同时新增convenience和required两个修饰初始化方法的关键词。convenience只是提供一种方便的初始化方法，必须通过调用同一个类中designated初始化方法来完成。required是强制子类重写父类中所修饰的初始化方法。

206. 谈谈对Objective-C和Swift 动态特性的理解。
参考答案：runtime其实就是Objective-C的动态机制。runtime执行的是编译后的代码，这时它可以动态加载对象、添加方法、修改属性、传递信息等等。具体过程是在Objective-C中对象调用方法时，如[self.tableview reload]，发生了两件事。

编译阶段，编译器（compiler）会把这句话翻译成objc_msgSend(self.tableview, @selector(reload))，把消息发送给self.tableview。

运行阶段，接收者self.tableview会响应这个消息，期间可能会直接执行、转发消息，也可能会找不到方法崩溃。

所以整个流程是编译器翻译 –> 给接收者发送消息 –> 接收者响应消息三个流程。

207. 在一个app中间有一个button，在你手触摸屏幕点击后，到这个button收到点击事件，中间发生了什么?
参考答案：响应链大概有以下几个步骤：
设备将touch到的UITouch和UIEvent对象打包, 放到当前活动的Application的事件队列中。
单例的UIApplication会从事件队列中取出触摸事件并传递给单例UIWindow。
UIWindow使用hitTest:withEvent:方法查找touch操作的所在的视图view。

RunLoop这边是这样的：
主线程的RunLoop被唤醒
通知Observer，处理Timer和Source 0
Springboard接受touch event之后转给App进程
RunLoop处理Source 1，Source1 就会触发回调，并调用_UIApplicationHandleEventQueue() 进行应用内部的分发。
RunLoop处理完毕进入睡眠，此前会释放旧的autorelease pool并新建一个autorelease pool

208. 介绍下内存的几大区域?

参考答案：1.栈区(stack) 由编译器自动分配并释放，存放函数的参数值，局部变量等。栈是系统数据结构，对应线程/进程是唯一的。优点是快速高效，缺点时有限制，数据不灵活。栈空间分静态分配 和动态分配两种。

2.堆区(heap) 由程序员分配和释放，如果程序员不释放，程序结束时，可能会由操作系统回收 ，比如在ios 中 alloc 都是存放在堆中。
优点是灵活方便，数据适应面广泛，但是效率有一定降低。虽然程序结束时所有的数据空间都会被释放回系统，但是精确的申请内存，释放内存匹配是良好程序的基本要素。

3.全局区(静态区) (static) 全局变量和静态变量的存储是放在一起的，初始化的全局变量和静态变量存放在一块区域，未初始化的全局变量和静态变量在相邻的另一块区域，程序结束后有系统释放。
4.文字常量区 存放常量字符串，程序结束后由系统释放；
5.代码区 存放函数的二进制代码

209. 你是如何组件化解耦的?

参考答案：「组件化」顾名思义就是把一个大的 App 拆成一个个小的组件，相互之间不直接引用。实现代码的高内聚低耦合，方便多人多团队开发！

一般需要解耦的项目都会多多少少出现，一下几个情况：

1.耦合比较严重（因为没有明确的约束，「组件」间引用的现象会比较多）

2.容易出现冲突（尤其是使用 Xib，还有就是 Xcode Project，虽说有脚本可以改善）

3.业务方的开发效率不够高（只关心自己的组件，却要编译整个项目，与其他不相干的代码糅合在一起）

210. runtime如何通过selector找到对应的IMP地址?

参考答案：类对象中有类方法和实例方法的列表，列表中记录着方法的名词、参数和实现，而selector本质就是方法名称，runtime通过这个方法名称就可以在列表中找到该方法对应的实现。

这里声明了一个指向struct objc_method_list指针的指针，可以包含类方法列表和实例方法列表
具体实现
在寻找IMP的地址时，runtime提供了两种方法
IMP class_getMethodImplementation(Class cls, SEL name);IMP method_getImplementation(Method m)

211. runloop内部实现逻辑?

参考答案：实际上 RunLoop 就是这样一个函数，其内部是一个 do-while 循环。当你调用 CFRunLoopRun() 时，线程就会一直停留在这个循环里；直到超时或被手动停止，该函数才会返回。

212. GCD执行原理?

参考答案：GCD有一个底层线程池，这个池中存放的是一个个的线程。之所以称为“池”，很容易理解出这个“池”中的线程是可以重用的，当一段时间后这个线程没有被调用，这个线程就会被销毁。注意：开多少条线程是由底层线程池决定的，池是系统自动来维护。

而程序员只关心的是向队列中添加任务，队列调度即可。如果队列中存放的是同步任务，则任务出队后，底层线程池中会提供一条线程供这个任务执行，任务执行完毕后这条线程再回到线程池。这样队列中的任务反复调度，因为是同步的，所以当我们用currentThread打印的时候，就是同一条线程。

如果队列中存放的是异步的任务，当任务出队后，底层线程池会提供一个线程供任务执行，因为是异步执行，队列中的任务不需等待当前任务执行完毕就可以调度下一个任务，这时底层线程池中会再次提供一个线程供第二个任务执行，执行完毕后再回到底层线程池中。

这样就对线程完成一个复用，而不需要每一个任务执行都开启新的线程，也就从而节约的系统的开销，提高了效率。

213. 怎么防止别人反编译你的app?

参考答案：1.本地数据加密：对UserDefaults，sqlite存储文件数据加密，保护帐号和关键信息

2.URL编码加密：对程序中出现的URL进行编码加密，防止URL被静态分析

3.网络传输数据加密：对客户端传输数据提供加密方案，有效防止通过网络接口的拦截获取数据

4.方法体，方法名高级混淆：对应用程序的方法名和方法体进行混淆，保证源码被逆向后无法解析代码

5.程序结构混排加密：对应用程序逻辑结构进行打乱混排，保证源码可读性降到最低

6.借助第三方APP加固：网易云易盾

214. 如何优化应用程序的启动速度?

参考答案： 启动过程中做的事情越少越好（尽可能将多个接口合并）
不在UI线程上作耗时的操作（数据的处理在子线程进行，处理完通知主线程刷新节目）
在合适的时机开始后台任务（例如在用户指引节目就可以开始准备加载的数据）
尽量减小包的大小

215. 如何优化应用程序的页面浏览速度?

参考答案： json的处理（iOS 自带的NSJSONSerialization，Jsonkit，SBJson）
数据的分页（后端数据多的话，就要分页返回，例如网易新闻，或者 微博记录）
数据压缩（大数据也可以压缩返回，减少流量，加快反应速度）
内容缓存（例如网易新闻的最新新闻列表都是要缓存到本地，从本地加载，可以缓存到内存，或者数据库，根据情况而定）
延时加载tab（比如app有5个tab，可以先加载第一个要显示的tab，其他的在显示时候加载，按需加载）
算法的优化（核心算法的优化，例如有些app 有个 联系人姓名用汉语拼音的首字母排序）

216. 如何优化应用程序的操作流畅度?

参考答案： Tableview 优化（tableview cell的加载优化）
ViewController加载优化（不同view之间的跳转，可以提前准备好数据）

217. 如何优化应用程序的客户端和服务器端的交互?

参考答案： 客户端尽量减少请求
服务端尽量做多的逻辑处理
服务器端和客户端采取推拉结合的方式（可以利用一些同步机制）
通信协议的优化。（减少报文的大小）
电量使用优化（尽量不要使用后台运行）

218. 如何优化应用程序的数据库?

参考答案： 数据库设计上面的重构
查询语句的优化
分库分表（数据太多的时候，可以分不同的表或者库）

219. 遇到tableView卡顿嘛？会造成卡顿的原因大致有哪些?

参考答案：可能造成tableView卡顿的原因有：
1.没有使用到cell的重用机制， 注册重用标识符。
2.避免cell的重新布局，cell的布局填充等操作 比较耗时，一般创建时就布局好。
3.提前计算并缓存cell的属性及内容，当我们创建cell的数据源方法时，编译器并不是先创建cell 再定cell的高度。
4.减少cell中控件的数量，不同风格的cell可以使用不用的重用标识符，初始化时添加控件，不适用的可以先隐藏。
5.不要使用ClearColor，无背景色，透明度也不要设置为0，渲染耗时比较长。
6.使用局部更新，如果只是更新某组的话，使用reloadSection进行局部更。
7.加载网络数据，下载图片，使用异步加载，并缓存。
8.少使用addView 给cell动态添加view。
9.按需加载cell，cell滚动很快时，只加载范围内的cell。
10.不要实现无用的代理方法，tableView只遵守两个协议。
13.预渲染图像。当新的图像出现时，仍然会有短暂的停顿现象。解决的办法就是在bitmap context里先将其画一遍，导出成UIImage对象，然后再绘制到屏幕；

220. timer准吗？谈谈你的看法？如果不准该怎样实现一个精确的Timer?

参考答案：不准，原因：
1、Timer加在main runloop中，模式是DefaultRunLoopMode，main负责所有主线程事件，例如UI界面的操作，复杂的运算，这样在同一个runloop中timer就会产生阻塞。
2、模式的改变。主线程的 RunLoop 里有两个预置的 Mode：kCFRunLoopDefaultMode 和 UITrackingRunLoopMode。

当你创建一个 Timer 并加到 DefaultMode 时，Timer 会得到重复回调，但此时滑动一个ScrollView时，RunLoop 会将 mode 切换为 TrackingRunLoopMode，这时 Timer 就不会被回调，并且也不会影响到滑动操作。所以就会影响到NSTimer不准的情况。

方法一：在主线程中进行Timer操作，但是将Timer实例加到main runloop的特定mode（模式）中。避免被复杂运算操作或者UI界面刷新所干扰。
方法二：使用mach内核级的函数可以使用mach_absolute_time()获取到CPU的tickcount的计数值，可以通过”mach_timebase_info”函数获取到纳秒级的精确度 。然后使用mach_wait_until(uint64_t deadline)函数，直到指定的时间之后，就可以执行指定任务了。

221. 编译过程做了哪些事情?

参考答案：1.C++,Objective C都是编译语言。编译语言在执行的时候，必须先通过编译器生成机器码，机器码可以直接在CPU上执行，所以执行效率较高。

iOS开发目前的常用语言是：Objective和Swift。二者都是编译语言，换句话说都是需要编译才能执行的。二者的编译都是依赖于Clang + LLVM.

iOS编译：不管是OC还是Swift，都是采用Clang作为编译器前端，LLVM(Low level vritual machine)作为编译器后端。
编译器前端的任务是进行：语法分析，语义分析，生成中间代码(intermediate representation)。
在这个过程中，会进行类型检查，如果发现错误或者警告会标注出来在哪一行。编译器后端会进行机器无关的代码优化，生成机器语言，并且进行机器相关的代码优化。LLVM机器码生成器会针对不同的架构，比如arm64等生成不同的机器码。编译各个.m文件，使用CompileC和clang命令。

222. 只有OC对象才需要进行内存管理的本质原因

参考答案：1.OC对象存放于堆中 2.非OC对象存在栈中（栈内存会被系统自动回收）

223. 请介绍一下isa指针，怎么用，用来干嘛的?

参考答案：每个Objective-C对象都有一个隐藏的数据结构，这个数据结构是Objective-C对象的第一个成员变量，它就是isa指针 ,这个指针是指向对象的真实类型，根据这个指针就能知道将来调用哪个类的方法

每个实例对象都有个isa指针, 他指向对象的类，而类对象里也有个isa指针, 指向meteClass(元类)。元类保存了类方法的方法列表。当类方法被调用时，先会从本身查找类方法的实现，如果没有，元类会向他父类查找该方法。

同时注意的是：元类（meteClass）也是类，它也是对象。元类也有isa指针, 它的isa指针最终指向的是一个根元类(root meteClass)，其实就是NSObject. 根元类的isa指针指向本身，这样形成了一个封闭的内循环。

224. @synthesize 和 @dynamic 分别有什么作用?

参考答案：@property 有两个对应的词,一个是@synthesize,一个是@dynamic。如果 @synthesize 和@dynamic 都没写,那么默认的就是@syntheszie var = _var; @synthesize 的语义是如果你没有手动实现 setter 方法和 getter 方法, 那么编译器会自动为你加上这两个方法

@dynamic 告诉编译器: 属性的 setter 与 getter 方法由用户自己实现, 不自动生 成(当然对于 readonly 的属性只需提供 getter 即可) 假如一个属性被声明为@dynamic var,然后你没有提供@setter 方法和@getter 方 法,编译的时候没问题,但是当程序运行到 instance.var = someVar,由于缺 setter 方法会导致程序崩溃;或者当运行到 someVar = instance.var 时,由于缺 getter 方 法同样会导致崩溃。编译时没问题,运行时才执行相应的方法,这就是所 谓的动态绑定

225. KVO 内部实现原理?

参考答案：KVO 是基于runtime机制实现的。当某个类的属性对象第一次被观察时, 系统就会在运行期动态地创建该类的一个派生类, 在这个派生类中重写基类中任何被观察属性的 setter 方法。派生类在被重写的 setter 方法内实现真正的通知机制。如果原类为 Person, 那么生成的派生类名为 NSKVONotifying_Person 每个类对象中都有一个 isa 指针指向当前类, 当一个类对象的第一次被观察, 那么系统会偷偷将 isa 指针指向动态生成的派生类, 从而在给被监控属性赋值时，执行的是派生类的 setter 方法。
键值观察通知依赖于 NSObject 的两个方法 : willChangeValueForKey: 和 didchangevluefoprkey:。
在一个被观察属性发生改变之前, willChangeValueForKey: 一定会被调用, 这就会记录旧的值。而当改变发生后, didChangeValueForKey: 会被调用, 继而 observeValueForKey:ofObject:change:context: 也会被调用。

226. 简述应用程序按Home键进入后台时的生命周期，以及从后台回到前台时的生命周期?

参考答案：App生命周期：

刚启动时:

application:didFinishLaunchingWithOptions:

applicationDidBecomeActive:

进入后台时

applicationWillResignActive:

applicationDidEnterBackground:

进入前台时

applicationWillEnterForeground:

applicationDidBecomeActive:

ViewController生命周期：

init/awakeFromNib > loadView > viewDidLoad > viewWillAppear > viewDidAppear:

227. NSOperationQueue相对于GCD来说有哪些优点?

参考答案：1.提供了在 GCD 中不那么容易复制的有用特性。

2.可以很方便的取消一个NSOperation的执行

3.可以更容易的添加任务的依赖关系

4.提供了任务的状态：isExecuteing, isFinished.

5.可以通过设置maxConcurrentOperationCount属性来控制并发任务的数量

228. runtime如何实现weak变量的自动置nil?

参考答案：runtime 对注册的类， 会进行布局，对于 weak 对象会放入一个 hash 表中。

用 weak 指向的对象内存地址作为 key，当此对象的引用计数为0的时候会 dealloc，假如 weak 指向的对象内存地址是a，那么就会以a为键， 在这个 weak 表中搜索，找到所有以a为键的 weak 对象，从而设置为 nil。

229. 常用设计模式及应用场景

参考答案：1、代理模式

应用场景：当一个类的某些功能需要由别的类来实现，但是又不确定具体会是哪个类实现。

优势：解耦合

2、观察者模式

应用场景：一个对象状态改变，通知正在对他进行观察的对象，这些对象根据各自要求做出相应的改变。

优势：解耦合

3、单例模式

应用场景：确保程序运行期某个类，只有一份实例，用于进行资源共享控制。

优势：使用简单，延时求值，易于跨模块

敏捷原则：单一职责原则

4、工厂模式

应用场景：工厂方式创建类的实例，多与proxy模式配合，创建可替换代理类。

优势：易于替换，面向抽象编程，application只与抽象工厂和易变类的共性抽象类发生调用关系。

230. UIButton的继承关系

参考答案：UIButton : UIControl : UIView : UIResponder : NSObject

231. 谈谈你对UITableView的优化和重用的理解。

参考答案：1.设置reuseIdentify，重用cell, 减少创建cell的内存开销
2.提前计算cell的高度并缓存，因为heightForRowAtIndexPath调用最频繁
3.不阻塞主线程，UITableView停止或者减速滑动结束的时候再进行异步加载图片（SDWebImage已经做到了）
4.有大量圆形头像的时候不要使用cornerRadius和masksToBounds设置，因为离屏渲染很消耗性能
5.减少subviews的层次，使用不透明视图，可以极大地提高渲染的速度：opaque = YES
6.如果只修改了某一行，就只更新单行
7.销毁的时候手动把代理设置为nil

232. 圆角图片如何优化?

参考答案：如果有大量圆角图片时，不要使用cornerRadius和masksToBounds，因为设置了这两个属性是离屏渲染性能消耗较大，可通过Core Graphics 画出圆角矩形（UIBezierPath画圆）

233. 什么是死锁，如何防止死锁?

参考答案：所谓死锁，通常指有两个线程A和B都卡住了，并等待对方完成某些操作。A不能完成是因为它在等待B完成。但B也不能完成，因为它在等待A完成。于是大家都完不成，就导致了死锁（DeadLock）

234. block为什么要修饰为copy?

参考答案：修饰为copy后会对block里面的对象retain一次，防止里面的对象提前释放

235. block循环引用，block分配在栈还是堆， 什么情况下block会从栈区复制到堆区?

参考答案：block有可能存在于三个地方：
_NSConcreteStackBlock：
只用到外部局部变量、成员属性变量，且没有强指针引用的block都是StackBlock。
StackBlock的生命周期由系统控制的，一旦返回之后，就被系统销毁了。

_NSConcreteMallocBlock：
有强指针引用或copy修饰的成员属性引用的block会被复制一份到堆中成为MallocBlock，没有强指针引用即销毁，生命周期由程序员控制

_NSConcreteGlobalBlock：
没有用到外界变量或只用到全局变量、静态变量的block为_NSConcreteGlobalBlock，生命周期从创建到应用程序结束。

236. OC如何调用js的代码；js调用OC的代码?

参考答案：shouldStartLoadWithRequest代理方法里面拦截 使用JavascriptCore框架里面的JSContext

237. 应用之间如何唤起?

参考答案：添加Schema [UIApplication sharedApplication] openURL 跳转

238. NSString为什么用copy修饰?

参考答案：一般情况下，我们都不希望字符串的值跟着原字符串的变化而变化，所以我们一般用copy来修饰NSString的属性。

239. JSPatch使用了iOS的什么原理?

参考答案：通过JS调用和改写OC方法最根本的原因是 Objective-C 是动态语言，OC上所有方法的调用/类的生成都通过 Objective-C Runtime 在运行时进行。

240. 使用drawRect有什么影响?

参考答案：drawRect方法依赖Core Graphics框架来进行自定义的绘制，但这种方法主要的缺点就是它处理touch事件的方式：每次按钮被点击后，都会用setNeddsDisplay进行强制重绘；而且不止一次，每次单点事件触发两次执行。

这样的话从性能的角度来说，对CPU和内存来说都是欠佳的。特别是如果在我们的界面上有多个这样的UIButton实例。

面试题第四篇

共有80题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

241. C和Objective-c如何混用

参考答案：1）obj-c的编译器处理后缀为m的文件时，可以识别obj-c和c的代码，处理mm文件可以识别obj-c,c,c++代码，但cpp文件必须只能用c/c++代码，而且cpp文件include的头文件中，也不能出现obj- c的代码，因为cpp只是cpp。
2) 在mm文件中混用cpp直接使用即可，所以obj-c混cpp不是问题
3）在cpp中混用obj- c其实就是使用obj-c编写的模块是我们想要的。
如果模块以类实现，那么要按照cpp class的标准写类的定义，头文件中不能出现obj-c的东西，包括#import cocoa的。

实现文件中，即类的实现代码中可以使用obj-c的东西，可以import,只是后缀是mm。
如果模块以函数实现，那么头文件要按 c的格式声明函数，实现文件中，c++函数内部可以用obj-c，但后缀还是mm或m。
总结：只要cpp文件和cpp include的文件中不包含obj-c的东西就可以用了，cpp混用obj-c的关键是使用接口，
而不能直接使用实现代码，实际上cpp混用的是 obj-c编译后的o文件，这个东西其实是无差别的，所以可以用。
obj-c的编译器支持cpp.

242. 谈谈你对静态链接库的理解?

参考答案：静态链接库就是.a文件，相当于java里的jar包，把一些类编译到一个包中，在不同的工程中如果导入此文件就可以使用里面的类，具体使用依然是#import “ xx.h”）。

243. 谈谈你对fmmpeg框架的理解?

参考答案：音视频编解码框架，内部使用UDP协议针对流媒体开发，内部开辟了六个端口来接受流媒体数据，完成快速接受之目的。

244. 谈谈你对fmdb框架的理解?

参考答案：数据库框架，对sqllite的数据操作进行了封装，使用户可以把精力都放在sql语句上面。

245. 什么是沙箱模型Sandbox?

参考答案：某个iphone工程进行文件操作有此工程对应的指定的位置，不能逾越。应该程序只能访问和操作位于沙箱中的资源。

246. 沙箱模型的有四个文件夹，分别是什么，永久数据存储一般放在什么位置，得到模拟器的路径的简单方式是什么.

参考答案： Documents 目录：您应该将所有de应用程序数据文件写入到这个目录下。这个目录用于存储用户数据或其它应该定期备份的信息。

AppName.app 目录：这是应用程序的程序包目录，包含应用程序的本身。由于应用程序必须经过签名，所以您在运行时不能对这个目录中的内容进行修改，否则可能会使应用程序无法启动。

Library 目录：这个目录下有两个子目录：Caches 和 Preferences。

Preferences 目录包含应用程序的偏好设置文件。您不应该直接创建偏好设置文件，而是应该使用NSUserDefaults类来取得和设置应用程序的偏好。

Caches 目录用于存放应用程序专用的支持文件，保存应用程序再次启动过程中需要的信息。

247. 你使用过json解析方式么，他们的底层是如何处理的你了解么?
参考答案：json解析的用法，用框架的用法简单介绍：

底层原理遍历字符串中的字符，最终根据格式规定的特殊字符，比如{}号，[]号, : 号 等进行区分，

{}号是一个字典的开始，[]号是一个数组的开始, : 号是字典的键和值的分水岭，最终乃是将json数据转化为字典，

字典中值可能是字典，数组，或字符串而已。

248. xml解析的原理是什么，你还用过其他解析方式么?
参考答案：NSXMLParser, 其他解析方式有自定义二进制解析，就是按字节去解析，电话会谈就是如此，
还可以是字符串之间用特殊符号连接的数据，将此数据用特殊符号可以分割成所用数据。

249. 你在开发大型项目的时候，如何进行内存泄露检测的?
参考答案：可以通过xcode的自带工具run---start with performance tool里有instruments下有个leaks工具，

启动此工具后，运行项目，工具里可以显示内存泄露的情况，双击可找到源码位置，可以帮助进行内存泄露的处理。

250. 让一个物体从界面中的一点运动到另外一点，有哪些方法?
参考答案：四种方式：
1. beginAnimation

2. 线程

3. NSTimer

4. 图层动画（路径）

251. 以下每行代码执行后，person对象的retain count分别是多少?
参考答案：Person *person = [[Person alloc] init]; count 1

[person retain]; retain count 2

[person release];retain count 1

[person release];retain count = 0

252. 在一个对象的方法里面:self.name = “object”;和name ＝”object”有什么不同吗?
参考答案：self.name = "object"会调用对象的setName()方法,会使object引用计数加1，name = "object"会直接把object赋值给当前对象的name 属性，引用计数不增加。

253. 谈谈您对@synthesize、@dynamic的理解

参考答案：@synthesize是系统自动生成getter和setter属性声明;@synthesize的意思是，除非开发人员已经做了，否则由编译器生成相应的代码，以满足属性声明；

@dynamic是开发者自已提供相应的属性声明,@dynamic意思是由开发人员提供相应的代码：对于只读属性需要提供getter，对于读写属性需要提供 setter 和getter。

254. 什么是设计模式？聊聊你所知道的设计模式。

参考答案：设计模式是为特定场景下的问题而定制的解决方案。特定场景指问题所在的重复出现的场景，问题指特定环境下你想达成的目标。同样的问题在不同的环境下会有不同的限制和挑战。定制的解决方案是指在特定环境下克服了问题的限制条件而达成目标的一种设计。

设计模式分为三种类型，共23种。
创建型模式：单例模式、抽象工厂模式、建造者模式、工厂模式、原型模式。
结构型模式：适配器模式、桥接模式、装饰模式、组合模式、外观模式、享元模式、代理模式。
行为型模式：模版方法模式、命令模式、迭代器模式、观察者模式、中介者模式、备忘录模式、解释器模式（Interpreter模式）、状态模式、策略模式、职责链模式(责任链模式)、访问者模式。

255. 请谈谈数组和链表的区别?

参考答案：首先从逻辑结构上说，两者都是数据结构的一种，但存在区别，数组是申请的一块连续的内存空间，并且是在编译阶段就要确定空间大小的，同时在运行阶段是不允许改变的，所以它不能够随着需要的改变而增加或减少空间大小，所以当数据量大的时候，有可能超出了已申请好的数组上限，产生数据越界，或者是数据量很小，对于没有使用的数组空间，造成内存浪费。链表则是动态申请的内存空间，并不像数组一样需要事先申请好大小，链表是现用现申请就OK，根据需求动态的申请或删除内存空间，对于的是增加或删除数据，所以比数组要灵活。

再从物理存储即内存分配上分析，数组是连续的内存，对于访问数据，可以通过下标直接读取，时间复杂度为O(1)，而添加删除数据就比较麻烦，需要移动操作数所在位置后的所有数据，时间复杂度为O（N）。链表是物理上非连续的内存空间，对于访问数据，需要从头便利整个链表直到找到要访问的数据，没有数组有效，但是在添加和删除数据方面，只需要知道操作位置的指针，很方便可以实现增删，教数组比较灵活有效率。所以综合以上，对于快速访问数据，不经常有添加删除操作的时候选择数组实现，而对于经常添加删除数据，对于访问没有很高要求的时候选择链表。

256. 谈谈你对顺序查找算法的看法?

参考答案：条件：无序或有序队列。
原理：按顺序比较每个元素，直到找到关键字为止。
时间复杂度：O(n)

257. 谈谈你对二分查找（折半查找）算法的看法?

参考答案：条件：有序数组
原理：查找过程从数组的中间元素开始，如果中间元素正好是要查找的元素，则搜素过程结束； 如果某一特定元素大于或者小于中间元素，则在数组大于或小于中间元素的那一半中查找，而且跟开始一样从中间元素开始比较。 如果在某一步骤数组为空，则代表找不到。 这种搜索算法每一次比较都使搜索范围缩小一半。
时间复杂度：O(logn)

258. 谈谈你对二叉排序树查找算法的看法?
参考答案：条件：先创建二叉排序树：

若它的左子树不空，则左子树上所有结点的值均小于它的根结点的值；
若它的右子树不空，则右子树上所有结点的值均大于它的根结点的值；
它的左、右子树也分别为二叉排序树。
原理：在二叉查找树b中查找x的过程为：

若b是空树，则搜索失败，否则：
若x等于b的根节点的数据域之值，则查找成功；否则：
若x小于b的根节点的数据域之值，则搜索左子树；否则：
查找右子树。

259. 谈谈你对哈希表法（散列表）的看法?
参考答案：条件：先创建哈希表（散列表）
原理：根据键值方式
(Key value)进行查找，通过散列函数，定位数据元素。
时间复杂度：几乎是O(1)，取决于产生冲突的多少。

260. 谈谈你对分块查找算法的看法?
参考答案：原理：将n个数据元素"按块有序"划分为m块（$m \leq n$）。 每一块中的结点不必有序，但块与块之间必须"按块有序"；
即第1块中任一元素的关键字都必须小于第2块中任一元素的关键字；
而第2块中任一元素又都必须小于第3块中的任一元素，然后使用二分查找及顺序查找。

261. UITableViewCell上有个UILabel，显示NSTimer实现的秒表时间，手指滚动cell过程中，label是否刷新，为什么?
参考答案：这是否刷新取决于timer加入到Run Loop中的Mode是什么。
Mode主要是用来指定事件在运行循环中的优先级的，分为：
NSDefaultRunLoopMode（kCFRunLoopDefaultMode）：默认，空闲状态
UITrackingRunLoopMode：ScrollView滑动时会切换到该Mode
UIInitializationRunLoopMode：run loop启动时，会切换到该Mode

苹果公开提供的Mode有两个：
NSDefaultRunLoopMode（kCFRunLoopDefaultMode）
NSRunLoopCommonModes（kCFRunLoopCommonModes）
在编程中：如果我们把一个NSTimer对象以NSDefaultRunLoopMode添加到主运行循环中的时候, ScrollView滚动过程中会因为Mode的切换，而导致NSTimer将不再被调度。当我们滚动的时候，也希望不调度，那就应该使用默认模式。但是，如果希望在滚动时，定时器也要回调，那就应该使用NSRunLoopCommonModes

262. 什么情况使用 weak 关键字，相比 assign 有什么不同?
参考答案：在 ARC 中,在有可能出现循环引用的时候,往往要通过让其中一端使用 weak 来解决,比如: delegate 代理属性。自身已经对它进行一次强引用,没有必要再强引用一次,此时也会使用 weak,自定义 IBOutlet 控件属性一般也使用 weak；当然，也可以使用strong。

不同点：
weak 此特质表明该属性定义了一种“非拥有关系” (nonowning relationship)。为这种属性设置新值时，设置方法既不保留新值，也不释放旧值。此特质同assign类似， 然而在属性所指的对象遭到摧毁时，属性值也会清空(nil out)。
而 assign 的“设置方法”只会执行针对“纯量类型” (scalar type，例如 CGFloat 或 NSlnteger 等)的简单赋值操作。
assign 可以用非 OC 对象,而 weak 必须用于 OC 对象。

263. 怎么用 copy 关键字?

参考答案：用途：

NSString、NSArray、NSDictionary 等等经常使用copy关键字，是因为他们有对应的可变类型：NSMutableString、NSMutableArray、NSMutableDictionary；

block 也经常使用 copy 关键字，block 使用 copy 是从 MRC 遗留下来的“传统”,在 MRC 中,方法内部的 block 是在栈区的,使用 copy 可以把它放到堆区.在 ARC 中写不写都行：对于 block 使用 copy 还是 strong 效果是一样的，但写上 copy 也无伤大雅，还能时刻提醒我们：编译器自动对 block 进行了 copy 操作。

如果不写 copy ，该类的调用者有可能会忘记或者根本不知道“编译器会自动对 block 进行了 copy 操作”，他们有可能会在调用之前自行拷贝属性值。这种操作多余而低效。

264. 这个写法会出什么问题： @property (copy) NSMutableArray *array;

参考答案：两个问题：

1、添加,删除,修改数组内的元素的时候,程序会因为找不到对应的方法而崩溃.因为 copy 就是复制一个不可变 NSArray 的对象；

2、使用了 atomic 属性会影响性能 ；

265. @protocol 和 category 中如何使用 @property

参考答案：在 protocol 中使用 property 只会生成 setter 和 getter 方法声明,我们使用属性的目的,是希望遵守我协议的对象能实现该属性

category 使用 @property 也是只会生成 setter 和 getter 方法的声明,如果真的需要给 category 增加属性的实现,需要借助于运行时的两个函数：

objc_setAssociatedObject
objc_getAssociatedObject

266. weak属性需要在dealloc中置nil么?

参考答案：不需要。

在ARC环境无论是强指针还是弱指针都无需在 dealloc 设置为 nil ， ARC 会自动帮我们处理。

即便是编译器不帮我们做这些，weak也不需要在 dealloc 中置nil。

267. ARC下，不显式指定任何属性关键字时，默认的关键字都有哪些?

参考答案：对应基本数据类型默认关键字是：

atomic,readwrite,assign

对于普通的 Objective-C 对象：

atomic,readwrite,strong

268. 用@property声明的NSString（或NSArray，NSDictionary）经常使用copy关键字，为什么？如果改用strong关键字，可能造成什么问题?

参考答案：因为父类指针可以指向子类对象,使用 copy 的目的是为了让本对象的属性不受外界影响,使用 copy 无论给我传入是一个可变对象还是不可对象,我本身持有的就是一个不可变的副本.

如果我们使用是 strong ,那么这个属性就有可能指向一个可变对象,如果这个可变对象在外部被修改了,那么会影响该属性。

copy 此特质所表达的所属关系与 strong 类似。然而设置方法并不保留新值，而是将其“拷贝” (copy)。 当属性类型为 NSString 时，经常用此特质来保护其封装性，因为传递给设置方法的新值有可能指向一个 NSMutableString 类的实例。这个类是 NSString 的子类，表示一种可修改其值的字符串，此时若是不拷贝字符串，那么设置完属性之后，字符串的值就可能会在对象不知情的情况下遭人更改。所以，这时就要拷贝一份“不可变” (immutable)的字符串，确保对象中的字符串值不会无意间变动。只要实现属性所用的对象是“可变的” (mutable)，就应该在设置新属性值时拷贝一份。

269. objc中向一个nil对象发送消息将会发生什么?
参考答案：在 Objective-C 中向 nil 发送消息是完全有效的，只是在运行时不会有任何作用:

如果一个方法返回值是一个对象，那么发送给nil的消息将返回0(nil)。

objc是动态语言，每个方法在运行时会被动态转为消息发送，即：objc_msgSend(receiver, selector)。

270. objc中向一个对象发送消息[obj foo]和objc_msgSend()函数之间有什么关系?
参考答案：该方法编译之后就是objc_msgSend()函数调用.

271. 什么时候会报unrecognized selector的异常?
参考答案：当调用该对象上某个方法，而该对象上没有实现这个方法的时候， 可以通过“消息转发”进行解决。

简单的流程如下:
objc是动态语言，每个方法在运行时会被动态转为消息发送，即：objc_msgSend(receiver, selector)。
objc在向一个对象发送消息时，runtime库会根据对象的isa指针找到该对象实际所属的类，然后在该类中的方法列表以及其父类方法列表中寻找方法运行，如果，在最顶层的父类中依然找不到相应的方法时，程序在运行时会挂掉并抛出异常unrecognized selector sent to XXX。

272. 一个objc对象如何进行内存布局? （考虑有父类的情况）
参考答案：所有父类的成员变量和自己的成员变量都会存放在该对象所对应的存储空间中.

每一个对象内部都有一个isa指针,指向他的类对象,类对象中存放着本对象的对象方法列表（对象能够接收的消息列表，保存在它所对应的类对象中），成员变量的列表和属性列表,它内部也有一个isa指针指向元对象(meta class),元对象内部存放的是类方法列表,类对象内部还有一个superclass的指针,指向他的父类对象。

273. 使用runtime Associate方法关联的对象，需要在主对象dealloc的时候释放么?
参考答案：无论在MRC下还是ARC下均不需要。

274. objc中的类方法和实例方法有什么本质区别和联系?
参考答案：类方法:
类方法是属于类对象的
类方法只能通过类对象调用
类方法中的self是类对象
类方法可以调用其他的类方法
类方法中不能访问成员变量
类方法中不能直接调用对象方法

实例方法:
实例方法是属于实例对象的
实例方法只能通过实例对象调用
实例方法中的self是实例对象
实例方法中可以访问成员变量
实例方法中直接调用实例方法
实例方法中也可以调用类方法(通过类名)

275. _objc_msgForward函数是做什么的，直接调用它将会发生什么?
参考答案：_objc_msgForward是 IMP 类型，用于消息转发的： 当向一个对象发送一条消息，但它并没有实现的时候，_objc_msgForward会尝试做消息转发。

276. runtime如何实现weak变量的自动置nil?
参考答案：runtime 对注册的类， 会进行布局，对于 weak 对象会放入一个 hash 表中。

用 weak 指向的对象内存地址作为 key，当此对象的引用计数为0的时候会 dealloc，假如 weak 指向的对象内存地址是a，那么就会以a为键， 在这个 weak 表中搜索，找到所有以a为键的 weak 对象，从而设置为 nil。

277. 能否向编译后得到的类中增加实例变量？能否向运行时创建的类中添加实例变量？为什么?
参考答案：不能向编译后得到的类中增加实例变量；

因为编译后的类已经注册在 runtime 中，类结构体中的 objc_ivar_list 实例变量的链表 和 instance_size 实例变量的内存大小已经确定，同时runtime 会调用 class_setIvarLayout 或 class_setWeakIvarLayout 来处理 strong weak 引用。所以不能向存在的类中添加实例变量；

运行时创建的类是可以添加实例变量，调用 class_addIvar 函数。但是得在调用 objc_allocateClassPair 之后，objc_registerClassPair 之前，原因同上，能向运行时创建的类中添加实例变量。

278. runloop和线程有什么关系?
参考答案：总的说来，Run loop，正如其名，loop表示某种循环，和run放在一起就表示一直在运行着的循环。

实际上，run loop和线程是紧密相连的，可以这样说run loop是为了线程而生，没有线程，它就没有存在的必要。Run loops是线程的基础架构部分， Cocoa 和 CoreFundation 都提供了 run loop 对象方便配置和管理线程的 run loop（以下都以 Cocoa 为例）。每个线程，包括程序的主线程（ main thread ）都有与之相应的 run loop 对象。

runloop和线程的关系：
主线程的run loop默认是启动的。
对其它线程来说，run loop默认是没有启动的，如果你需要更多的线程交互则可以手动配置和启动，如果线程只是去执行一个长时间的已确定的任务则不需要。
在任何一个 Cocoa 程序的线程中，都可以通过以下代码来获取到当前线程的 run loop：
NSRunLoop *runloop = [NSRunLoop currentRunLoop];

279. runloop的mode作用是什么?
参考答案：model 主要是用来指定事件在运行循环中的优先级的，分为：

NSDefaultRunLoopMode（kCFRunLoopDefaultMode）：默认，空闲状态
UITrackingRunLoopMode：ScrollView滑动时
UIInitializationRunLoopMode：启动时
NSRunLoopCommonModes（kCFRunLoopCommonModes）：Mode集合

苹果公开提供的 Mode 有两个：
NSDefaultRunLoopMode（kCFRunLoopDefaultMode）
NSRunLoopCommonModes（kCFRunLoopCommonModes）

280. objc使用什么机制管理对象内存?
参考答案：通过 retainCount 的机制来决定对象是否需要释放。
每次 runloop 的时候，都会检查对象的 retainCount，如果retainCount 为 0，说明该对象没有地方需要继续使用了，可以释放掉了。

281. ARC通过什么方式帮助开发者管理内存?

参考答案：ARC相对于MRC，不是在编译时添加retain/release/autorelease这么简单。应该是编译期和运行期两部分共同帮助开发者管理内存。

在编译期，ARC用的是更底层的C接口实现的retain/release/autorelease，这样做性能更好，也是为什么不能在ARC环境下手动retain/release/autorelease，同时对同一上下文的同一对象的成对retain/release操作进行优化（即忽略掉不必要的操作）；

ARC也包含运行期组件，这个地方做的优化比较复杂，但也不能被忽略。

282. 不手动指定autoreleasepool的前提下，一个autorealese对象在什么时刻释放？（比如在一个vc的viewDidLoad中创建）

参考答案：分两种情况：手动干预释放时机、系统自动去释放。

手动干预释放时机：指定autoreleasepool，当前作用域大括号结束时释放。

系统自动去释放：不手动指定autoreleasepool，Autorelease对象出了作用域之后，会被添加到最近一次创建的自动释放池中，并会在当前的 runloop 迭代结束时释放。

283. BAD_ACCESS在什么情况下出现?

参考答案：访问了悬垂指针，比如对一个已经释放的对象执行了release、访问已经释放对象的成员变量或者发消息、死循环。

284. 苹果是如何实现autoreleasepool的?

参考答案：autoreleasepool 以一个队列数组的形式实现,主要通过下列三个函数完成.

objc_autoreleasepoolPush
objc_autoreleasepoolPop
objc_autorelease

看函数名就可以知道，对 autorelease 分别执行 push，和 pop 操作。销毁对象时执行release操作。

285. 使用block时什么情况会发生引用循环，如何解决?

参考答案：一个对象中强引用了block，在block中又强引用了该对象，就会发射循环引用。

解决方法是将该对象，使用__weak或者__block修饰符修饰之后再在block中使用。

id weak weakSelf = self;
或者 weak __typeof(&*self)weakSelf = self该方法可以设置宏
id __block weakSelf = self;
或者将其中一方强制置空：
xxx = nil。

286. 在block内如何修改block外部变量?

参考答案：默认情况下，在block中访问的外部变量是复制过去的，即：写操作不对原变量生效。但是你可以加上__block 来让其写操作生效，示例代码如下:

__block int a = 0;

287. 使用系统的某些block api（如UIView的block版本写动画时），是否也考虑引用循环问题?
参考答案：系统的某些block api中，UIView的block版本写动画时不需要考虑。

288. GCD的队列（dispatch_queue_t）分哪两种类型?
参考答案：串行队列Serial Dispatch Queue和并行队列Concurrent Dispatch Queue。

289. 如何用GCD同步若干个异步调用？（如根据若干个url异步加载多张图片，然后在都下载完成后合成一张整图）
参考答案：使用Dispatch Group追加block到Global Group Queue,这些block如果全部执行完毕，就会执行Main Dispatch Queue中的结束处理的block。

290. dispatch_barrier_async的作用是什么?
参考答案：在并行队列中，为了保持某些任务的顺序，需要等待一些任务完成后才能继续进行，使用 barrier 来等待之前任务完成，避免数据竞争等问题。

dispatch_barrier_async 函数会等待追加到Concurrent Dispatch Queue并行队列中的操作全部执行完之后，然后再执行 dispatch_barrier_async 函数追加的处理，等 dispatch_barrier_async 追加的处理执行结束之后，Concurrent Dispatch Queue才恢复之前的动作继续执行。

291. 若一个类有实例变量 NSString *_foo ，调用setValue:forKey:时，可以以foo还是 _foo 作为key?
参考答案：都可以。

292. KVC和KVO的keyPath一定是属性么?
参考答案：KVC 支持实例变量，KVO 只能手动支持手动设定实例变量的KVO实现监听。

293. IBOutlet连出来的视图属性为什么可以被设置成weak?
参考答案：因为既然有外链那么视图在xib或者storyboard中肯定存在，视图已经对它有一个强引用了。

使用storyboard（xib不行）创建的vc，会有一个叫_topLevelObjectsToKeepAliveFromStoryboard的私有数组强引用所有top level的对象，所以这时即便outlet声明成weak也没关系。

294. IB中User Defined Runtime Attributes如何使用?
参考答案：它能够通过KVC的方式配置一些你在interface builder 中不能配置的属性。当你希望在IB中作尽可能多得事情，这个特性能够帮助你编写更加轻量级的viewcontroller

295. lldb（gdb）有哪些常用的调试命令?
参考答案：breakpoint：设置断点定位到某一个函数
n：断点指针下一步
po：打印对象

296. 使用工厂模式最主要的好处是什么?
参考答案：工厂模式的最大好处是增加了创建对象时的封装层次。如果你使用工厂来创建对象，之后你可以使用更高级和更高性能的实现来替换原始的产品实现或类，这不需要在调用层做任何修改。

297. UIscrollVew用到了什么设计模式?
参考答案：组合模式composition，所有的container view都用了这个模式。
观察者模式observer，所有的UIResponder都用了这个模式。
模板(Template)模式，所有datasource和delegate接口都是模板模式的典型应用。

298. 描述一种在Swift中出现循环引用的情况，并说明怎么解决。
参考答案：循环引用出现在当两个实例对象相互拥有强引用关系的时候，这会造成内存泄露，原因是这两个对像都不会被释放。只要一个对象被另一个对象强引用，那么该对象就不能被释放，由于强引用的存在，每个对象都会保持对方存在。

解决这个问题的方法是，用weak或者unowned引用代替其中一个的强引用，来打破循环引用。

299. 闭包是引用类型吗?
参考答案：闭包是引用类型。

如果一个闭包被分配给一个变量,这个变量复制给另一个变量,那么他们引用的是同一个闭包，他们的捕捉列表也会被复制。

300. 泛型类，泛型结构体，泛型协议的功能特性都包含在Swift中吗?
参考答案：Swift 包含泛型类，泛型结构体特性。泛型可以在类、结构体、枚举、全局函数或者方法中使用。

泛型协议是通过typealias部分实现的。typealias不是一个泛型类型,它只是一个占位符的名字。它通常是作为关联类型被引用，只有协议被一个类型引用的时候它才被定义。

301. 对一个optional变量拆包有多少种方法？并在安全方面进行评价。
参考答案：强制拆包！ 操作符——不安全
隐式拆包变量声明——大多数情况下不安全
可选绑定——安全
自判断链接（optional chaining）——安全
nil coalescing 运算符（空值合并运算符）——安全
Swift 2.0 的新特性 guard 语句——安全
Swift 2.0 的新特性optional pattern（可选模式） ——安全

302. 哪些情况下你不得不使用隐式拆包？说明原因。
参考答案：对optional变量使用隐式拆包最常见的原因如下：

对象属性在初始化的时候不能nil,否则不能被初始化。典型的例子是Interface Builder outlet类型的属性，它总是在它的拥有者初始化之后再初始化。在这种特定的情况下，假设它在Interface Builder中被正确的配置——outlet被使用之前，保证它不为nil。

解决强引用的循环问题——当两个实例对象相互引用，并且对引用的实例对象的值要求不能为nil时候。在这种情况下，引用的一方可以标记为unowned,另一方使用隐式拆包。

建议：除非必要，不要对option类型使用隐式拆包。使用不当会增加运行时崩溃的可能性。在某些情况下,崩溃可能是有意的行为,但有更好的方法来达到相同的结果,例如,通过使用fatalError( )函数

303. 什么是泛型？泛型是用来解决什么问题的?
参考答案：泛型是用来使类型和算法安全的工作的一种类型。在Swift中，在函数和数据结构中都可以使用泛型，例如类、结构体和枚举。

泛型一般是用来解决代码复用的问题。常见的一种情况是，你有一个函数，它带有一个参数，参数类型是A，然而当参数类型改变成B的时候，你不得不复制这个函数。

304. 在Swfit中,什么时候用结构体，什么时候用类?
参考答案：一直都有这样的争论：到底是用类的做法优于用结构体，还是用结构体的做法优于类。函数式编程倾向于值类型，面向对象编程更喜欢类。
在Swift 中，类和结构体有许多不同的特性。下面是两者不同的总结：

类支持继承，结构体不支持。
类是引用类型，结构体是值类型。
并没有通用的规则决定结构体和类哪一个更好用。一般的建议是使用最小的工具来完成你的目标，但是有一个好的经验是多使用结构体，除非你用了继承和引用语义。
注意：在运行时，结构体的在性能方面更优于类，原因是结构体的方法调用是静态绑定，而类的方法调用是动态实现的。这就是尽可能得使用结构体代替类的又一个好的原因。

305. 什么是optional类型，它是用来解决什么问题的?
参考答案：optional类型被用来表示任何类型的变量都可以表示缺少值。在Objective-C中，引用类型的变量是可以缺少值得，并且使用nil作为缺少值。基本的数据类型如int 或者float没有这种功能。

Swift用optional扩展了在基本数据类型和引用类型中缺少值的概念。一个optional类型的变量，在任何时候都可以保存一个值或者为nil。

306. 简要描述观察者模式?
参考答案：观察者模式（Observer）是指一个或多个对象对另一个对象进行观察，当被观察对象发生变化时，观察者可以直接或间接地得到通知，从而能自动地更新观察者的数据，或者进行一些操作。 具体到iOS的开发中，实现观察者模式常用的方式有KVO和Notification两种。

两者的不同在于，KVO是被观察者主动向观察者发送消息；Notification是被观察者向NotificationCenter发送消息，再由NotificationCenter post通知到每个注册的观察者。

307. 分辨率的计算单位是什么?
参考答案：描述分辨率的单位有：（dpi点每英寸）、lpi（线每英寸）和ppi（像素每英寸）。

308. 设备状态栏（Device Status Bar）是什么？高度如何？是否透明？在手机通话或者导航状态下，它是如何显示的?
参考答案：ios上状态栏 就是指的最上面的20像素高的部分
状态栏分前后两部分：

前景部分：就是指的显示电池、时间等部分；
背景部分：就是显示黑色或者图片的背景部分；

309. 选项卡（Tab Bar）和工具栏（Toolbar）分别是什么？两者之间有何共同点和不同点?
参考答案： Tab Bar:提供一种机制来显示一个选项卡位于屏幕底部的工具栏和支持显示和选择的选项卡栏项(UITabBarItem的实例)。您可以使用UITabBarDelegate指定自定义视图选项卡栏项。

Toolbar:提供一种机制来显示一个工具栏在屏幕的底部和支持显示和选择的工具栏项目(UIToolbarItem的实例)。您可以使用UIToolbarDelegate指定自定义工具栏项目视图。

310. 表视图（Table View）是什么？集合视图（Collection View）又是什么?
参考答案：UITableView：
UITableView 继承于UIScrollVIew,可以滚动.
UITableView 的每一条数据对应的单元格叫做Cell, UITableViewCell的一个对象,继承于UIView.
UITableView 可以分区显示, 每一个分区称为section, 每一行称为row, 编号都从0开始.

UICollectionView：
UICollectionView是一种类似于UITableView但又比UITableView功能更强大、更灵活的视图，这是源于它将UICollectionView对cell的布局交给了UICollectionViewLayout，而且允许用户自定义layout来进行布局。

311. 选取器视图（Picker View）适合存放哪类内容？What sort of content would be appropriate to place in a picker view?
参考答案：UIPickerView控件在给用户选择某些特定的数据时经常使用到，比如选择性别 / 国家 / 地区 等等。

312. 应该在什么情况下使用标签、文本域和文本视图？When are a label, text field and text view appropriate?
参考答案：UILabel：显示只读文本。
最最基本的显示文字的，支持单行或多行，wrapped，truncated，styled。UILabel是UIView的一个子类，而UIView又是UIResponder的子类，就使得UILabel支持：position定位, color着色, mask掩码，access访问，触摸touch。
UITextView：可编辑文本，提供换行功能，支持多行时滚动。允许垂直方向的滚动。文字是可选的，可以拷贝，可以选中后调用语音读出来。

UITextField：可编辑本文，但是无法换行，只能在一行显示；
当点击键盘上的return时会收到一个事件做一些事情。
当内容太长一行没有完全显示时：提供水平方向的字符的定位seeking。
主要用于：简短的文字输入。

313. 什么是分段控件？What does a segmented control do?
参考答案：SegmentedControl又被称作分段控制器，是iOS开发中经常用到的一个UI控件。

通常是在单视图中使用，不做多视图之间的切换。实现视图中不同显示的快速切换，每一个分段表示一个不同的显示,这些显示往往是相关的,所谓的相关，可以理解成，功能一样，但是属性类别有差异。

比较常用的还有，在一个视图中，不同的分段控制tableView加载不同的数据源。
它通常在整个屏幕的上部，不是必然，但大部分情况下是这样用。
一般是3到5个分段，超过5个的话每个分段的大小对于用户触碰的体验会很差。
任意时刻，只有一个分段是激活状态的。有点像单选按钮。

314. 什么是模态视图？What is a modal view?
参考答案：模态视图和普通视图最主要的区别就是模态视图显示的时候，不能对其他视图进行操作。

315. iOS应用图标是指什么？请尽可能详细地描述一下。

参考答案：每个app都需要一个漂亮的icon，依靠icon的外观来评判/确定一款app的质量、目的以及可靠性的用户并不在少数。

当你思考app cion设计的时候，需要记住几件事。当你开始创造它的时候，要看看App Icon以获得详细的指南和说明。

Icon是app品牌重要的一部分，icon设计为你提供了一个向用户讲述故事，和用户建立情感联系的机会。

最好的app icon是独一无二的，是井然有序的，是迷人的。

不管在什么背景上，不管是什么尺寸，App icon的外观都应该是很好看的。细节会让大尺寸的icon变得丰富充实，但会让小尺寸的icon看起来杂乱不堪。

316. 请解释一下启动画面（Launch Images）

参考答案：什么是启动画面?

启动应用程序后，进入主功能界面前会有一张图片或一段动画效果，停留数秒钟后消失。这张图片或这段动画效果我们称之为应用的启动画面。由于启动画面在每次打开应用时都会出现，并且停留很短的时间，就像闪现的效果一样，所以也有人把启动画面称之为闪屏。

为什么需要启动画面?

从用户体验的角度考虑，产品应当尽可能快的让用户进入程序并进行功能操作。如此看来，启动画面就是多余的存在。但是，应用程序本质上是由多种代码命令和数据组合而成的。启动应用程序的过程实质上是运行代码和数据读取的过程，而这个过程是需要一定时间的。与其给用户看代码运行的过程，不如给用户展示图片，这也是用户体验。

317. 自动布局（Auto Layout）的作用是什么？请概括一下它是如何运行的。

参考答案：在iOS6中引入的Auto Layout，使你能够灵活的创建布局。相比于使用 autoresizingmasks 和手动布局的方式来布局程序的用户界面，使用Auto Layout将会是一个更好的选择。

你可以使用Auto Layout来给视图添加约束并且定义他们之间的关系。这个关系可以是视图和它的父视图之间的，和兄弟视图之间的关系，甚至是和它自己的关系。

相对于之间的直接明确的设置视图的frame，Auto Layout可以使用约束来设定两个视图之间的间距和相对位置。Auto Layout使用这些约束条件来计算用户视图上的元素在程序运行时的位置。

你必须给视图设置足够的约束条件以防止位置的模糊不清。另外需注意的是，设置过多的约束条件也会导致冲突甚至程序的崩溃。

318. 为什么map函数必不可少？该在什么情况下使用它?

参考答案：map是一种基本的数据结构，它提供了很好一对一的关系，能够非常方便的进行查找，插入，删除，排序等等。

根据行为、效率、排序策略、保存对象的生命周期和判定“键”等价的策略等不同，可以衍生为各种map，例如HashMap, TreeMap, LinkedHashMap, WeakHashMap, IdentityHashMap。

319. 请解释一下UserDefaults。就你而言，你会如何在磁盘中对数组对象进行序列化?

参考答案：UserDefaults是系统自带的一个轻量级数据本地化的一个方法，其本质是一个单例。用户名、密码、网址等都适合使用UserDefaults来储存。对于自定义的对象需要对其归档才能储存。

320. AppDelegate扮演着什么样的角色?

参考答案：创建应用程序之后之后，默认有AppDelegate.swift文件。 AppDelegate为整个应用的一个代理，提供程序启动、退出等类似监控的接口，控制着应用的生命周期。

面试题第五篇

共有80题

扫码免费下载互动教程，手把手教学模式！或在苹果商店(App Store)搜索互动教程。

321. Metal是什么?

参考答案：在iOS 8里，苹果发布了一个新的接口叫做Metal，它是一个支持GPU加速3D绘图的API。

Metal和OpenGL ES相似，它也是一个底层API，负责和3D绘图硬件交互。它们之间的不同在于，Metal不是跨平台的。与之相反的，它设计的在苹果硬件上运行得极其高效，与OpenGL ES相比，它提供了更快的速度和更低的开销。

322. 请描述SpriteKit和SceneKit的作用。

参考答案：SpriteKit在IOS和OS X中均可使用，它使用主机设备提供的图形硬件来高帧速率合成2D图片，SpriteKit支持很多不同的内容，包括：
无纹理或者有纹理的举行（精灵）、文本、任意基于CGPath的形状和视频。

SpriteKit同时支持裁剪以及其他的一些效果：你可以将这些效果应用于所有或者一部分内容，你可以在每一帧中活动（animate）或者改变这些元素（elememt）。你也可以附加物理主体（physics bodies）到这些元素中，使得他们支持推动和碰撞。

因为SpriteKit支持丰富的渲染结构并接管了所有向OpenGL发出绘画指令的的底层工作（low-level work），你可以将你的重心放在解决高层次的设计问题以及伟大的游戏交互设计上。

SceneKit可以同时在IOS和OS X下工作, 是用来构建3D场景的框架，且可以与Core Animation和SpriteKit无缝交互。在SceneKit中可以直接引入COLLADA行业标准文件制作好的3D模型或场景。

323. 请概括描述一下Core Audio是什么框架?

参考答案：Core Audio 是iOS和 MAC 的关于数字音频处理的基础，它提供应用程序用来处理音频的一组软件框架，所有关于IOS音频开发的接口都是由Core Audio来提供或者经过它提供的接口来进行封装的。按照官方的说法是集播放、音频处理、录制为一体的专业技术，通过它我们的程序可以同时录制，播放一个或者多个音频流，自动适应耳机，蓝牙耳机等硬件，响应各种电话中断，静音，震动等，甚至提供3D效果的音乐播放。

324. 请概括描述一下Core Data是什么框架?

参考答案：Core Data框架提供了对象-关系映射(ORM)的功能，即能够将OC对象转化成数据，保存在SQLite3数据库文件中，也能够将保存在数据库中的数据还原成OC对象。
在此数据操作期间，不需要编写任何SQL语句。

325. 请概括描述一下Core Location是什么框架?

参考答案：Core Location Framework的用途是使用户通过移动设备来获取定位信息和方向信息，还有你的范围，当用户走过某些范围边界就能马上监控到，在iOS上，甚至可以连同beacon联动来确定周边信息。

326. iBeacons是什么?

参考答案：iBeacons是苹果在WWDC 2013上有意无意透漏出来的一项重要功能，通过低功耗蓝牙(BluetoothLowEnergy)技术进行十分精确的微定位和室内导航，据悉其定位精度可以以厘米计算。

实现iBeacons精准的微定位功能除了需要运行iOS 7且支持BLE的设备外，还需要在室内、店内或者其他公共环境中部署iBeacon基站。当用户走进信号覆盖区域内时，用户就会收到相关的提醒和询问。以梅西百货为例，当用户走到商场某个店面附近时，安装了相应app的用户就会收到由iBeacons基站发出的产品信息或者打折信息。此外，美国职棒大联盟（MLB）也已经测试使用了iBeacon 技术，苹果更是在254间Apple Store 里应用了iBeacon 技术。

对于开发者来说，可以创建一个更加具有交互性的博物馆应用，当用户在博物馆内随意行走时，通过信息提醒用户某些特别的展览。技术还可以用作室内导航，比如在地铁站或者机场这些GPS信号不大好的地方更好地引导用户。

327. Game Center针对iOS游戏有哪些功能?

参考答案：简单而言， Game Center提供了以下机制：
用户管理和验证：你的游戏不用再考虑用户注册，登录，密码找回，存储分数等繁琐的功能实现。Game Center为你的游戏提供了一个本地玩家（local player），这个对象为所有的游戏共享，你只需要调用即可；
排行榜：你只需要在 iTunesConnect中指定排行方式就可以轻松拥有一个或者多个排行榜;
记录成就：你的游戏需要定义里程碑，调用Game Center就可以保存进度；
挑战：Game Center提供了让玩家挑战其他玩家的机制。

328. HealthKit是什么?

参考答案：HealthKit是Apple公司在推出iOS 8 系统时一块推出的关于健康信息的框架。如果iPhone手机系统升级到iOS8之后就会发现多了一个健康-app,这就是Apple提供的一个记录用户健康信息的app,可以用它来分享健康和健身数据。还可以指定数据的来源，比如我们自己创建一个app,在我们的app中使用了HealthKit框架之后只要经过用户的认证，就可以在我们的app之中给健康分享数据或者从健康中获取数据。

HealthKit可以与健身设备一起工作，iPhone手机自身可以监控步数信息，会自动导入步数信息。但是其他信息或者设备需要配套的应该才能获取到数据并导入到HealthKit中并在健康中显示。

329. 请描述一下你的iOS应用开发流程。

参考答案：注册APP ID
注册成为苹果开发者
创建开发证书、发布证书
开发
真机调试
打包和发布
等待审核
上线

330. iOS是如何提高安全性，保护用户隐私信息的?

参考答案：要求使用https链接并验证证书的正确性
对用户敏感数据 使用密钥进行加密
对传输的内容进行加密

为了获取所在地（Location）数据，iOS8增加了一条新的While In Use许可项。这就意味着在未活跃的状态下，若有应用想访问所在地，就会弹出和该应用有关的警告信息。用户很方便地就可以取消该许可。

根据儿童网络隐私保护法案（COPPA），iOS8上的儿童应用必须有隐私政策，并作为“大致的规则”，不会收集任何形式的可识别出某个人的有关信息。这就意味着，儿童应用上不应有广告行为，也不应要求获取所在地、联系人、图像等要素。

Keychain和Touch ID在iPhone5s中就存在在安全区域中了，而iOS8的这些特性却加强了很多隐私设置，尤其是当设备使用Touch ID允许应用进入时，只能接触到设备本人，这会使用户的银行应用相对安全些。

331. performSelector:withObject:afterDelay: 内部大概是怎么实现的，有什么注意事项么?

参考答案：创建一个定时器,时间结束后系统会使用runtime通过方法名称(Selector本质就是方法名称)去方法列表中找到对应的方法实现并调用方法。

注意事项:
调用performSelector:withObject:afterDelay:方法时,先判断希望调用的方法是否存在respondsToSelector:
这个方法是异步方法,必须在主线程调用,在子线程调用永远不会调用到想调用的方法

332. 为什么当 Core Animation 完成时，layer 又会恢复到原先的状态?
参考答案：因为这些产生的动画只是假象,并没有对layer进行改变。

那么为什么会这样呢,这里要讲一下图层树里的呈现树.呈现树实际上是模型图层的复制,但是它的属性值表示了当前外观效果,动画的过程实际上只是修改了呈现树,并没有对图层的属性进行改变,所以在动画结束以后图层会恢复到原先状态

333. 当 TableView 的 Cell 改变时，如何让这些改变以动画的形式呈现?
参考答案：[tableView beginUpdates];
[tableView endUpdates];
让我们来重点关注这行代码:[tableView beginUpdates]; 文档中对这行代码的解释为让TableView产生插入，删除或重新加载cell。

和[tableView reloadata]是完全不一样的风格，reloadData这个方法会让tableView整体重新加载，相当于是作用在tableView上，而beginUpdates只是作用在cell上，这样看上就就会以动画的形式显示Cell的变化。

334. Push Notification 是如何工作的?
参考答案：用户的设备与苹果APNS服务器形成一个长连接,用户设备会发送uuid和Bundle idenidentifier给苹果服务器,苹果服务器会加密生成一个deviceToken给用户设备,然后设备会将deviceToken发送给APP的服务器,服务器会将deviceToken存进他们的数据库。

如果有人发送消息给某一个用户,服务器端就会去查询该用户的deviceToken,然后将deviceToken和要发送的信息发送给苹果服务器,苹果服务器通过deviceToken找到该用户的设备并将消息推送到设备上。

335. 在Objective-C中，如何为 Class 定义一个对外只读对内可读写的属性?
参考答案：在头文件中将属性定义为readonly, 在.m文件中将属性重新定义为readwrite。

336. 哪些途径可以让 ViewController 瘦下来?
参考答案：把 Data Source 和其他 Protocols 分离出来(将UITableView或者UICollectionView的代码提取出来放在其他类中)
将业务逻辑移到 Model 中(和模型有关的逻辑全部在model中写)
把网络请求逻辑移到 Model 层(网络请求依靠模型)
把 View 代码移到 View 层(自定义View)

337. 有哪些常见的 Crash 场景?
参考答案：访问了僵尸对象
访问野指针
访问了不存在的方法
数组越界
在定时器下一次回调前将定时器释放

338. autorelease 对象在什么情况下会被释放?
参考答案：手动干预释放就是指定autoreleasepool,当前作用域大括号结束就立即释放

系统自动去释放:不手动指定autoreleasepool,Autorelease对象会在当前的 runloop 迭代结束时释放

kCFRunLoopEntry(1):第一次进入会自动创建一个autorelease

kCFRunLoopBeforeWaiting(32):进入休眠状态前会自动销毁一个autorelease,然后重新创建一个新的autorelease

kCFRunLoopExit(128):退出runloop时会自动销毁最后一个创建的autorelease

339. 当系统出现内存警告时会发生什么?
参考答案：会将不在当前窗口上的view暂时移除
如果放任内存警告,最终会导致软件强制被系统关闭

340. Toll-Free Bridging 是什么？什么情况下会使用?
参考答案：Toll-Free Bridging用于在Foundation对象与Core Foundation对象之间交换数据,俗称桥接

在ARC环境下,Foundation对象转成 Core Foundation对象

使用__bridge桥接以后ARC会自动管理2个对象
使用__bridge_retained桥接需要手动释放Core Foundation对象
在ARC环境下, Core Foundation对象转成 Foundation对象

使用__bridge桥接,如果Core Foundation对象被释放,Foundation对象也同时不能使用了,需要手动管理Core Foundation对象 使用__bridge_transfer桥接,系统会自动管理2个对象

341. 为什么 NotificationCenter 要 removeObserver? 如何实现自动 remove?
参考答案：如果不移除的话,万一注册通知的类被销毁以后又发了通知,程序会崩溃.因为向野指针发送了消息
实现自动remove:通过自释放机制,通过动态属性将remove转移给第三者,解除耦合,达到自动实现remove

342. block 实现原理
参考答案：Objective-C是对C语言的扩展，block的实现是基于指针和函数指针。
从计算语言的发展，最早的goto，高级语言的指针，到面向对象语言的block，从机器的思维，一步步接近人的思维，以方便开发人员更为高效、直接的描述出现实的逻辑（需求）。

343. __block在arc和非arc下含义一样吗?
参考答案：是不一样的。

在MRC中__block variable在block中使用是不會retain的
但是ARC中__block則是會Retain的。
取而代之的是用__weak或是__unsafe_unretained來更精確的描述weak reference的目的
其中前者只能在iOS5之後可以使用，但是比較好 (該物件release之後，此pointer會自動設成nil)
而後者是ARC的環境下為了相容4.x的解決方案。

344. 使用nonatomic一定是线程安全的吗?
参考答案：不是的。

atomic原子操作，系统会为setter方法加锁。 具体使用 @synchronized(self){//code }
nonatomic不会为setter方法加锁。
atomic：线程安全，需要消耗大量系统资源来为属性加锁

345. UIScrollView 大概是如何实现的，它是如何捕捉、响应手势的?
参考答案：UIScrollView在滚动过程当中，其实是在修改原点坐标。当手指触摸后, scroll view会暂时拦截触摸事件, 使用一个计时器。

假如在计时器到点后没有发生手指移动事件，那么 scroll view 发送 tracking events 到被点击的 subview。

假如在计时器到点前发生了移动事件，那么 scroll view 取消 tracking 自己发生滚动。

346. Objective-C 中，meta-class 指的是什么?
参考答案：meta-class 是 Class 对象的类, 为这个Class类存储类方法, 当一个类发送消息时,就去那个类对应的meta-class中查找那个消息。

每个Class都有不同的meta-class, 所有的meta-class都使用基类的meta-class(假如类继承NSObject,那么他所对应的meta-class也是NSObject) 作为他们的类。

347. iOS 的签名机制大概是怎样的?
参考答案：签名机制：

先将应用内容通过摘要算法，得到摘要
再用私钥对摘要进行加密得到密文
将源文本、密文、和私钥对应的公钥一并发布

验证流程：
查看公钥是否是私钥方的
然后用公钥对密文进行解密得到摘要
将APP用同样的摘要算法得到摘要，两个摘要进行比对，如果相等那么一切正常
以上过程只要有一步出问题就视为无效。

348. 谈谈控制器View的加载过程?
参考答案：当程序访问了控制器的View属性时会先判断控制器的View是否存在，
如果存在就直接返回已经存在的View；
如果不存在，就会先调用loadView这个方法；
如果控制器的loadView方法实现了，就会按照loadView方法加载自定义的View；
如果控制器的loadView方法没有实现就会判断storyboard是否存在；
如果storyboard存在就会按照storyboard加载控制器的View；
如果storyboard不存在，就会创建一个空视图返回。

349. 设计一套大文件（如上百M的视频）下载方案。
参考答案：大文件下载注意事项

若不对下载的文件进行转存，会造成内存消耗急剧升高，甚至耗尽内存资源，造成程序终止。
在文件下载过程中通常会出现中途停止的状况，若不做处理，就要重新开始下载，浪费流量。
大文件下载的解决方案

对下载文件进行处理，每下载一点数据，就将数据写到磁盘中（通常是沙盒中），避免在内存累积数据（NSURLConnection下载）
使用NSFileHandle类实现写数据
使用NSOutputStream类实现写数据
当下载任务终止时，记录任务终止时的位置信息，以便下次开始继续下载

350. 什么是谓词?
参考答案：谓词是通过NSPredicate，是通过给定的逻辑条件作为约束条件，完成对数据的筛选。

351. 用过Core Data 或者 SQLite吗？读写是分线程的吗？遇到过死锁没？如何解决的?
参考答案：使用NSLock加锁。

352. 什么是NSManagedObject模型?
参考答案：NSManagedObject是NSObject的子类 ，也是coredata的重要组成部分，它是一个通用的类,实现了core data 模型层所需的基本功能，用户可通过子类化NSManagedObject，建立自己的数据模型。

353. MD5和Base64的区别是什么，各自场景是什么?
参考答案：md5： 用户密码存储 文件校验
base64： 公开的代码加密 url加密

354. SDWebImage原理
参考答案：从内存中（字典）找图片（当这个图片在本次程序加载过），找到直接使用；
从沙盒中找，找到直接使用，缓存到内存。
从网络上获取，使用，缓存到内存，缓存到沙盒。

355. 单例模式的作用?
参考答案：单例模式的作用是解决“应用中只有一个实例”的一类问题。
dispatch_once函数是由GCD提供的，它的作用是在整个应用程序生命周期中只执行一次代码块。
dispatch_once_t是由GCD提供的结构体，使用时需要将GCD地址传给dispatch_once函数。
dispatch_once函数能够记录该代码是否被调用过。
dispatch_once函数不仅意味着代码仅会被执行一次，而且还意味着此运行还是线程同步的。
也就是说，当我们使用dispatch_once函数时，就不需要再使用诸如@synchronized之类的语句。

356. 程序中有哪些常见的单例?
参考答案：UIApplication、NSUserDefaults、NSNotificationCenter、NSFileManager、NSBundle

357. 请问ViewController 的 loadView,、viewDidLoad,、viewDidUnload 分别是在什么时候调用的?
参考答案：viewDidLoad在view从nib文件初始化时调用，loadView在controller的view为nil时调用。
此方法在编程实现view时调用,view控制器默认会注册memory warning notification,当view controller的任何view没有用的时候，viewDidUnload会被调用，在这里实现将retain的view release,如果是retain的IBOutlet view 属性则不要在这里release,IBOutlet会负责release 。

358. Quatrz 2D的绘图功能的三个核心概念是什么并简述其作用。
参考答案：上下文：主要用于描述图形写入哪里；

路径：是在图层上绘制的内容；

状态：用于保存配置变换的值、填充和轮廓， alpha 值等。

359. 自定义实现UITabbarController的原理
参考答案：运用字典，点击五个按钮的一个可以从字典里选择一个控制器对象，将其View显示到主控制器视图上。

360. 请谈谈你对沙盒的理解。
参考答案：每个iOS应用都被限制在“沙盒”中，沙盒相当于一个加了仅主人可见权限的文件夹，及时在应用程序安装过程中，系统为每个单独的应用程序生成它的主目录和一些关键的子目录。苹果对沙盒有几条限制:

应用程序在自己的沙盒中运作，但是不能访问任何其他应用程序的沙盒；
应用之间不能共享数据，沙盒里的文件不能被复制到其他
应用程序的文件夹中，也不能把其他应用文件夹复制到沙盒中；
苹果禁止任何读写沙盒以外的文件，禁止应用程序将内容写到沙盒以外的文件夹中；

361. LayoutSubViews在什么时候被调用?
参考答案：当View本身的frame改变时，会调用这个方法。

362. 二叉搜索树的概念，时间复杂度多少?
参考答案：采用二叉树链表作为存储结构，每个左节点均小于父节点，每个右节点均大于父节点
时间复杂度：O(log2(n))

363. 如何制作一个静态库/动态库，他们的区别是什么?
参考答案：无论是动态库还是静态库都是区分真机和模拟器的，需要lipo命令进行合并。
静态库编译静态库文件装入程序空间，动态库是文件动态装入内存。
动态库执行到相关函数才会被调用，节省空间。

364. isMemberOfClass 和 isKindOfClass 联系与区别是什么?
参考答案：联系：两者都能检测一个对象是否是某个类的成员
区别：isKindOfClass 不仅用来确定一个对象是否是一个类的成员,也可以用来确定一个对象是否派生自该类的类的成员 ,而isMemberOfClass 只能做到第一点。

365. 用你熟悉的语音,编程实现Fibonacci数列:int F(intn);

参考答案：Fibonacci数列递推式F(n) = F(n-1) +F(n-2)

F(1) = 1
F(2) = 2
F(3) = 3
F(4) = 5
F(5) = 8

```
int F(int n){
    if(n == 1){
        return1;
    }
    return f(n-1)+f(n-2);
}
```

366. 给定两个排好序的数组A,B,请写一个函数,从中找出他们的公共元素:findCommon(A,B)，并列举其他可能的查找方法,越多越好

参考答案：例如:Array A = [1, 3, 5, 6, 9]，Array B = [2, 3, 6, 8, 10]，返回结果= [3, 6]。

```
void FindCommon(int* a, int* b, int n){
    int i = 0;int j = 0 ;
    while(i < n && j < n){
        if (a[i] < b[j])
            ++i;
        else if(a[i] == b[j]){
            cout << a[i] << endl ;
            ++i ;
            ++j ;
        }
        else
            ++j ;
}
```

367. KVO的实现原理?

参考答案：当指定的对象的属性被修改了，允许对象接收到通知的机制。

368. 如何给一个对象的私有属性赋值?

参考答案：利用KVC即键值编码来给对象的私有属性赋值.

369. block的本质是什么?为啥在block里面更改外面变量的值,要给外面的变量加_block修饰,加_block修饰的原理是什么?

参考答案：(1) block本质是一个数据类型,多用于参数传递,代替代理方法, (有多个参数需要传递或者多个代理方法需要实现还是推荐使用代理方法),少用于当做返回值传递. block是一个OC对象,它的功能是保存代码片段,预先准备好代码,并在需要的时候执行.

(2)因为使用block代码块可能会引起内部循坏引用,所以应在block定义前加上修饰

370. NSURLSession在什么情况下回存在循环引用的问题,怎么解决?

参考答案：(1)在使用NSURLSession签订其代理的时候会存在循环引用问题，因为其代理是retain强引用

(2)解决方案

（1）在下载完成后取消NSURLSession会话并释放Session，赋值为nil。

（2）再视图将要消失时也执行同样的操作。为了防止没有下载完成就跳转控制器。

371. 如何自己实现GET缓存?

参考答案：答:1.使用GET请求数据

2.iOS系统SDK已经做好了缓存。需要的仅仅是设置下内存缓存大小、磁盘缓存大小、以及缓存路径,代码如下

```
NSURLCache *urlCache = [[NSURLCache alloc] initWithMemoryCapacity:4 * 1024 * 1024 diskCapacity:20 * 1024 * 1024 diskPath:nil];

[NSURLCache setSharedURLCache:urlCache];
```

372. 在使用SQLite过程中,如果多条线程同时操作同一数据库会造成什么问题,怎么解决?

参考答案：(1)容易造成系统崩溃

(2)解决方案：开启第3种串行模式，使用一个类（单例方式）操作数据库。

373. 在GCD中线程之间怎么通信?请举例说明。

参考答案:

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT,0), ^{        //下载图片
    UIImage *image = nil;
    dispatch_async(dispatch_get_main_queue(),^{
        //回到主线程
});
```

374. NSThread的线程之间如何通信？请举例说明。

参考答案:

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT,0), ^{        //下载图片
    UIImage *image = nil;
            [selfperformSelector:@selector(settingImage:)        onThread:[NSThread        mainThread]withObject:image waitUntilDone:YES modes:nil];
}
```

375. 网络图片处理问题中怎么解决一个相同的网络地址重复请求的问题?

参考答案：利用字典图片地址为key，下载操作为value。

376. 自动释放池底层怎么实现?

参考答案：自动释放池以栈的形式实现:当你创建一个新的自动释放池时,它将被添加到栈顶.当一个对象收到发送autorelease消息时,它被添加到当前线程的处于栈顶的自动释放池中,当自动释放池被回收时,它们从栈中被删除,并且会给池子里面所有的对象都会做一次release操作。

377. 不用中间变量,用两种方法交换A和B的值

参考答案：A = A+B;

B = A - B;

A = A - B;

378. 用预处理指令#define声明一个常数,用以表明一年中有多少秒?(忽略闰年问题)

参考答案：#define second 365*24*60*60

379. 如何生成同时支持多个架构(simulator,arm7,arm64)的通用静态库?

参考答案：ValidArchitectures设置为：armv7｜armv7s｜arm64｜i386｜x86_64；

Architectures设置不变（或根据你需要）:armv7｜arm64；

然后分别选择iOS设备和模拟器进行编译，最后找到相关的.a进行合包，使用lipo -create真机库.a的路径模拟器库.a的的路径－output合成库的名字.a；

380. @class用途

参考答案：@class一般用于头文件中声明某个类的实例变量的时候用到.它只是声明,至于内部的实现是没有告诉编译器的。

381. 请用Objective-C代码如何判断某个对象obj是否支持某个method.

参考答案：if ([srespondsToSelector:@selector(print:)]) {

[s print:@"支持这个方法"];

}

382. 3升的杯子一个,5升的杯子一个,杯子的形状不规则,问怎么才能得到4升的水,水无限多.

参考答案：先将5升的杯子倒满,然后把5升的杯子中的水倒入3升的杯子,倒满后5升的杯子剩下2升。

再把3升杯子中的水倒掉,把5升的杯子中剩余的2升水倒入3升的杯子中,然后把5升的杯子倒满。

再用5升的杯子中的水给3升的杯子添满,则5升的杯子中剩余4升的水.

383. 如何处理多个网络请求并发的情况?

参考答案：了解(并发)当有多个线程在操作时,如果系统只有一个CPU,则它根本不可能真正同时进行一个以上的线程，它只能把CPU运行时间划分成若干个时间段,再将时间段分配给各个线程执行，在一个时间段的线程代码运行时，其它线程处于挂起状。.这种方式我们称之为并发(Concurrent)。

遇到这种情况建议使用第三方的网络库。比如AFNetworking。也可以通过GCD和NSOperationQueue来控制并发。

384. 简单介绍一下KVC和KVO，他们都可以应用在哪些场景?
参考答案：KVO:键值监听,观察某一属性的方法

KVC:键值编码,是一种间接访问对象的属性。

385. 什么是Plist文件?
参考答案：Plist文件通常用于储存用户设置，也可以用于存储捆绑的信息，该功能在旧式的Mac OS中是由资源分支提供的。

Plist主要有Core Fundation类型构成,也可以将这些类型放入NSDictionary和NSArray以便后膛更复杂的数据类型

386. What is CoreFoundation framework,andwhat is Foundation frame-work。什么是框架的CoreFoundation，什么是Foundation框架?
参考答案：Core Foundation框架(CoreFoundation.framework)是一组C语言接口，它们为iOS应用程序提供基本数据管理和服务功能。

Foundation—基础框架。框架中包含了很多开发中常用的数据类型，如结构体，枚举，类等，是其他ios框架的基础。

如果要想使用foundation框架中的数据类型，那么包含它的主头文件就可以了。

387. Swift的内存管理是怎样的?
参考答案：Swift 使用自动引用计数（Automatic Reference Counting, ARC）来简化内存管理。

388. Swift 支持面向过程编程吗?
参考答案：它采用了 Objective-C 的命名参数以及动态对象模型，可以无缝对接到现有的 Cocoa 框架，并且可以兼容 Objective-C 代码，支持面向过程编程和面向对象编程

389. 举例说明Swift里面有哪些是 Objective-C中没有的?
参考答案：Swift引入了在Objective-C中没有的一些高级数据类型，例如tuples（元组），可以使你创建和传递一组数值。

Swift还引入了可选项类型（Optionals），用于处理变量值不存在的情况。可选项的意思有两种：一是变量是存在的，例如等于X，二是变量值根本不存在。Optionals类似于Objective-C中指向nil的指针，但是适用于所有的数据类型，而非仅仅局限于类，Optionals 相比于Objective-C中nil指针更加安全和简明，并且也是Swift诸多最强大功能的核心。

390. Swift 是一门安全语言吗?
参考答案：Swift是一门类型安全的语言，Optionals就是代表。Swift能帮助你在类型安全的环境下工作，如果你的代码中需要使用String类型，Swift的安全机制能阻止你错误的将Int值传递过来，这使你在开发阶段就能及时发现并修正问题。

391. Swift中如何定义变量和常量?

参考答案：使用let来声明常量，使用var来声明变量。

392. 用Swift定义一个数组和字典?

参考答案：let emptyArray = String[]()
let emptyDictionary = Dictionary()

393. 为什么要在变量类型后面加个问号?

参考答案：用来标记这个变量的值是可选的。

394. 给一个数组，要求写一个函数，交换数组中的两个元素?

参考答案：func swap(_ nums: inout [T], _ p: Int, _ q: Int) {
(nums[p], nums[q]) = (nums[q], nums[p])
}

395. Swift 中定义常量和 Objective-C 中定义常量有什么区别?

参考答案：一般人会觉得没有差别，因为写出来好像也确实没差别。

OC是这样定义常量的：

const int number = 0;
Swift 是这样定义常量的：

let number = 0
首先第一个区别，OC中用 const 来表示常量，而 Swift 中用 let 来判断是不是常量。

上面的区别更进一步说，OC中 const 表明的常量类型和数值是在 compilation time 时确定的；而 Swift 中 let 只是表明常量（只能赋值一次），其类型和值既可以是静态的，也可以是一个动态的计算方法，它们在 runtime 时确定的。

396. Swift 到底是面向对象还是函数式的编程语言?

参考答案：说 Swift 是 Object-oriented，是因为 Swift 支持类的封装、继承、和多态，从这点上来看与 Java 这类纯面向对象的语言几乎毫无差别。

说 Swift 是函数式编程语言，是因为 Swift 支持 map, reduce, filter, flatmap 这类去除中间状态、数学函数式的方法，更加强调运算结果而不是中间过程

397. 可以使用扩展添加存储属性吗?

参考答案：这是不可能的。扩展可以为已经存在的数据类型添加新的行为，但是不允许改变类型本身或它的接口。

如果您添加了存储的属性，您需要额外的内存来存储新的值。扩展不能完成这样的任务。

398. 闭包是值类型还是引用类型的?

参考答案：闭包是引用类型。如果一个闭包被分配给一个变量，该变量被复制到另一个变量，它们实际是引用的相同一个闭包并且它里面的参数列表也同样会被复制。

399. Optional(可选型)是怎么实现的?

参考答案：在Objective-C中并没有Optional类型, 只有nil,并且nil只能用于表示对象类型无值,

并不能用于基础类型(int, float),枚举和结构体,基础类型需要返回类似NSNotFound的特殊值来表示无值,

所以在Swift中定义了Optinal类型来表示各种类型的无值状态,并规定了nil不能用于非可选的常量和变量,只能用于Optinal类型.

400. 谈谈你对[1,2,3].map{$0 * 2}代码的理解

参考答案：[1, 2, 3] 是阵列的語法，意義為字面上表示為 Array 的陣列。
Trailing Closures。如果一個函數的最後一個參數是閉包時，則可以不寫 ()。
省略閉包參數的類型聲明，其被推斷為 Int。
$0 表明是陣列中的每一個元素，省略了閉包參數的命名。
省略了閉包回傳值類型的聲明，其被推斷為 Int。
若閉包只有一行，則可以省略 return。