1.

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC中类的声明关键字为（）。（选一项）** | |
|  | **a)** | @import |
|  | **b)** | @interface |
|  | **c)** | @implemention |
|  | **d)** | @protocole |
| 标准答案 | | b |
| 考核知识点 | | 类的定义 |
| 难度 | | 1 |
| 所属章节 | | 0204 |
| 认知层次 | | 1 |

2.

|  |  |  |
| --- | --- | --- |
| **二)** | **现有Person类声明和Person的子类Student，代码如下：**  **@interface Person : NSObject**  **-(void) walk;**  **@end**  **@implementation Person**  **-(void) walk**  **{**  **NSLog(@”Person is walking…”);**  **}**  **@end**  **@interface Student : Person**  **-(void) studentWalk;**  **@end**  **@implementation Student**  **-(void) studentWalk**  **{**  **(1)**  **}**  **end**  **在下列选项中，那些代码可以填在(1)处实现调用Person类中的walk方法，并输出“Person is walking…”（）（选两项）。** | |
|  | **a)** | walk(); |
|  | **b)** | [super walk]; |
|  | **c)** | [self walk]; |
|  | **d)** | self.walk(); |
| 标准答案 | | bc |
| 考核知识点 | | 继承、super、self关键字 |
| 难度 | | 2 |
| 所属章节 | | 0303 |
| 认知层次 | | 4 |

3.

|  |  |  |
| --- | --- | --- |
| **二)** | **现有如下集合，能准确删除“张三”的代码是（）。（选一项）**  **NSMutableDictionary \* peoples=[NSMutableDictionary dictionaryWithObjectsAndKeys:@"张三",@"左护法",@"李四",@"右使",@"唐sir",@"老大哥", nil];** | |
|  | **a)** | [peoples removeObjectForKey:@"左护法"]; |
|  | **b)** | [peoples removeObject:@"张三"]; |
|  | **c)** | [peoples removeObjectAtIndex:0]; |
|  | **d)** | [peoples removeAllObjects]; |
| 标准答案 | | a |
| 考核知识点 | | 字典集合 |
| 难度 | | 1 |
| 所属章节 | | 0303 |
| 认知层次 | | 3 |

4.

|  |  |  |
| --- | --- | --- |
| **二)** | **查看下列代码，能正确输出number的选项是（）。（选一项）**  **@interface MyClass : NSObject**  **{**  **int number;**  **}**  **+(void) printNumber;**  **@end**  **@implementation MyClass**  **+(void) printNumber**  **{**  **number =10;**  **NSLog(@"%d",number);**  **}**  **@end** | |
|  | **a)** | [MyClass printNumber]; |
|  | **b)** | MyClass \* obj = [[MyClass alloc] init];  [obj printNumber]; |
|  | **c)** | printNumber(); |
|  | **d)** | 编译错误，无法执行。 |
| 标准答案 | | d |
| 考核知识点 | | 类方法 |
| 难度 | | 3 |
| 所属章节 | | 0202 |
| 认知层次 | | 4 |

5.

|  |  |  |
| --- | --- | --- |
| **二)** | **现有自定义类Student，下列选项中Student类的便利构造器编写正确的是（）。（选一项）** | |
|  | **a)** | -(id) initWithName:(NSString \*) newName{  Student \* stu = [[Student alloc] init];  stu.Name = newName;  return stu;  } |
|  | **b)** | +(id) studentWithName:(NSString \*) newName{  {  Student \* stu = [[Student alloc] init];  stu.Name = newName;  return stu;  } |
|  | **c)** | +(id) studentWithName:(NSString \*) newName{  {  Student \* stu = [[Student alloc] init];  stu.Name = newName;  [stu release];  return stu;  } |
|  | **d)** | +(id) studentWithName:(NSString \*) newName{  {  Student \* stu = [[Student alloc] init];  stu.Name = newName;  [stu autorelease];  return stu;  } |
| 标准答案 | | d |
| 考核知识点 | | 便利构造器 |
| 难度 | | 2 |
| 所属章节 | | 0305 |
| 认知层次 | | 2 |

6.

|  |  |  |
| --- | --- | --- |
| **二)** | **阅读下列代码，回答问题**  **@interface Employee**  **@property (copy,nonatomic) NSString \* name;**  **@end**  **@implementation Employee**  **@synthesize name=\_name;**  **@end**  **#import "Employee.h"**  **int main (int argc, const char \* argv[])**  **{**  **@autoreleasepool {**  **Employee \* emp = [[Employee alloc] init];**  **}**  **}**  **return 0;**  **}**  **下列选项中能正常为emp的name属性赋值的代码有（）（选三项）** | |
|  | **a)** | emp.name=@”李嘉诚”; |
|  | **b)** | emp->name=@”李开复”; |
|  | **c)** | [emp setValue:@”柳传志” forKey:@”name”]; |
|  | **d)** | [emp setName:@”黄光裕”]; |
| 标准答案 | | acd |
| 考核知识点 | | 属性，kvc |
| 难度 | | 3 |
| 所属章节 | | 0204 |
| 认知层次 | | 4 |

7.

|  |  |  |
| --- | --- | --- |
| **二)** | **关于协议描述正确的有（）。（选三项）** | |
|  | **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。 |
| 标准答案 | | abc |
| 考核知识点 | | 协议 |
| 难度 | | 2 |
| 所属章节 | | 0603 |
| 认知层次 | | 2 |

8.

|  |  |  |
| --- | --- | --- |
| **二)** | **NSString 和 NSMutableString的关系和区别描述不正确的是（）。（选一项）** | |
|  | **a)** | NSString是不可变字符串，即本身内容不能被改变。 |
|  | **b)** | NSMutableString是可变字符串，即本身内容可修改。 |
|  | **c)** | NSMutableString是NSString的子类。 |
|  | **d)** | NSString和NSMutableString没有任何关系。 |
| 标准答案 | | d |
| 考核知识点 | | 字符串 |
| 难度 | | 2 |
| 所属章节 | | 0701 |
| 认知层次 | | 2 |

9．

|  |  |  |
| --- | --- | --- |
| **二)** | **阅读NSTimer的便利构造器，回答问题：**  **+(NSTimer \*)scheduledTimerWithTimeInterval:(NSTimeInterval)ti target:(id)aTarget selector:(SEL)aSelector userInfo:(id)userInfo repeats:(BOOL)yesOrNo;**  **对于此方法参数描述错误的是（）。（选一项）** | |
|  | **a)** | ti：说明在多长时间后执行定时器，单位是秒。 |
|  | **b)** | target：调用对象，即那个对象触发了定时器。 |
|  | **c)** | aSelector：触发定时器的要执行的方法。 |
|  | **d)** | repeats：定时器 |
| 标准答案 | | b |
| 考核知识点 | | NSTimer |
| 难度 | | 1 |
| 所属章节 | | 0804 |
| 认知层次 | | 2 |

10.

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC中，能够从文件中读取内容的类有（）。（选两项）** | |
|  | **a)** | NSString |
|  | **b)** | NSData |
|  | **c)** | NSDate |
|  | **d)** | NSTimer |
| 标准答案 | | ab |
| 考核知识点 | | 文件操作 |
| 难度 | | 1 |
| 所属章节 | | 0904 |
| 认知层次 | | 2 |

11.

|  |  |  |
| --- | --- | --- |
| **二)** | **现要求声明一个集合，并将字符串添加到集合中，编写了如下代码，其中错误的是第几行（）。（选一项）**  NSString \* aString = [[NSString alloc] initWith**String**:@"ibokan"]; --（1）  [aString autorelease]; --（2）  NSArray \* arr = [NSArray array]; --（3）  [arr addObject:aString]; --（4）  NSLog(@"string = %lu",[aString retainCount]); | |
|  | **a)** | (1) |
|  | **b)** | (2) |
|  | **c)** | (3) |
|  | **d)** | (4) |
|  | **e)** | 无错误 |
| 标准答案 | | d |
| 考核知识点 | | 数组集合 |
| 难度 | | 2 |
| 所属章节 | | 0703 |
| 认知层次 | | 2 |

12.

|  |  |  |
| --- | --- | --- |
| **二)** | **对于下面代码说法正确的是（）。（选一项）**  **@property (copy,nonatomic,readonly)NSString \* calValue；** | |
|  | **a)** | 当对calvalue赋值时，将产生一个NSString副本。 |
|  | **b)** | 此属性可保证多线程安全。 |
|  | **c)** | 此属性在合成时可以生成设置器方法。 |
|  | **d)** | 此属性只能读取，不能赋值。 |
| 标准答案 | | b |
| 考核知识点 | | 类的定义 |
| 难度 | | 3 |
| 所属章节 | | 0204 |
| 认知层次 | | 4 |

13.

|  |  |  |
| --- | --- | --- |
| **二)** | **下列代码正确的输出结果是（）。（选一项）**  **NSString \* urlStr = @”www.ibokanwisdom.com”;**  **NSRange range = NSMakeRange(4,6);**  **NSString \*prefix = [urlStr substringWithRange:range];**  **NSLog(@”%@”,prefix);** | |
|  | **a)** | ib |
|  | **b)** | www.ib |
|  | **c)** | ibokan |
|  | **d)** | okan |
| 标准答案 | | c |
| 考核知识点 | | 字符串方法 |
| 难度 | | 2 |
| 所属章节 | | 0702 |
| 认知层次 | | 3 |

14.

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC中扩展类的方法的形式有（）。（选三项）** | |
|  | **a)** | 继承 |
|  | **b)** | 多态 |
|  | **c)** | 类目 |
|  | **d)** | 延展 |
| 标准答案 | | acd |
| 考核知识点 | | 类、继承、类目、延展 |
| 难度 | | 2 |
| 所属章节 | | 0601、0602、0603 |
| 认知层次 | | 3 |

15.

|  |  |  |
| --- | --- | --- |
| **二)** | **关于NSSet描述正确的有（）。（选二项）** | |
|  | **a)** | NSSet可以存放任何数据类型的数据。 |
|  | **b)** | NSSet存放数据是无序的。 |
|  | **c)** | NSSet可以通过下标获取数据。 |
|  | **d)** | NSSet可以通过anyObject方法返回一个元素。 |
| 标准答案 | | bd |
| 考核知识点 | | NSSet |
| 难度 | | 2 |
| 所属章节 | | 0705 |
| 认知层次 | | 2 |

16

|  |  |  |
| --- | --- | --- |
| **二)** | **关于集合描述不正确的是（）。（选一项）** | |
|  | **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、集这三种集合存储数据的方式不同。 |
|  | **d)** | 所有集合都这能保存id类型数据。 |
| 标准答案 | | a |
| 考核知识点 | | 集合 |
| 难度 | | 1 |
| 所属章节 | | 0703、0704、0705 |
| 认知层次 | | 2 |

17

|  |  |  |
| --- | --- | --- |
| **二)** | **关于类目的说法正确的有（）（选二项）** | |
|  | **a)** | 类目可以为所有的类添加新方法。 |
|  | **b)** | 类目中添加的方法比原类中定义的方法级别要低。 |
|  | **c)** | 定义类目需要使用关键字category。 |
|  | **d)** | 类目不能随意使用。 |
| 标准答案 | | ad |
| 考核知识点 | | 类目 |
| 难度 | | 2 |
| 所属章节 | | 0601 |
| 认知层次 | | 2 |

18

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC的KVO实现中，下列描述正确的是（）。（选一项）** | |
|  | **a)** | 使用KVO可以在观察者的属性改变时，发送消息给被观察者。 |
|  | **b)** | 添加观察者的关键代码为：  [被观察者 addObserver:观察者 forKeyPath:属性名字符串 options:观察选项 context:nil]; |
|  | **c)** | 添加观察者的关键代码为：  [观察者 addObserver:被观察者 forKeyPath:属性名字符串 options:观察选项 context:nil]; |
| 标准答案 | | b |
| 考核知识点 | | KVO |
| 难度 | | 2 |
| 所属章节 | | 0403 |
| 认知层次 | | 3 |

19

|  |  |  |
| --- | --- | --- |
| **二)** | **关于KVC的描述正确的是（）。（选二项）** | |
|  | **a)** | KVC是指“键-值-编码”。 |
|  | **b)** | 是一种间接访问对象的属性的机制。 |
|  | **c)** | 只能访问对象的属性。 |
|  | **d)** | 当对象的属性值改变时，我们能收到一个通知。 |
| 标准答案 | | ab |
| 考核知识点 | | KVC |
| 难度 | | 2 |
| 所属章节 | | 0401 |
| 认知层次 | | 2 |

20

|  |  |  |
| --- | --- | --- |
| **二)** | **内存管理的关键字描述错误的是（）。（选一项）** | |
|  | **a)** | retaincount是指对象的引用计数。 |
|  | **b)** | retain关键字可以增大对象的引用计数。 |
|  | **c)** | release和autorelease可以减小对象的引用计数。 |
|  | **d)** | retaincount为0时，对象内存将被回收。 |
| 标准答案 | | c |
| 考核知识点 | | 内存管理 |
| 难度 | | 2 |
| 所属章节 | | 0302 |
| 认知层次 | | 2 |

21

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC中，类的定义包括两部分，这两部分是（）。（选两项）** | |
|  | **a)** | @import |
|  | **b)** | @interface |
|  | **c)** | @implementation |
|  | **d)** | @protocol |
| 标准答案 | | bc |
| 考核知识点 | | 类的定义 |
| 难度 | | 1 |
| 所属章节 | | 0203 |
| 认知层次 | | 1 |

22

|  |  |  |
| --- | --- | --- |
| **二)** | **下列选项中全部属于对象的是（）。（选一项）** | |
|  | **a)** | 我的白色iPhone4s;Mac Mini |
|  | **b)** | 狗;老虎 |
|  | **c)** | 邻居家的猫咪“贝贝”;我的弟弟“张三” |
|  | **d)** | 保时捷;犀利哥 |
| 标准答案 | | c |
| 考核知识点 | | 类和对象 |
| 难度 | | 1 |
| 所属章节 | | 0202 |
| 认知层次 | | 2 |

23

|  |  |  |
| --- | --- | --- |
| **二)** | **以下对类的描述正确的是（）。（选三项）** | |
|  | **a)** | 类的接口部分可以包含实例变量、属性和方法 |
|  | **b)** | 可以把声明和实现都放到.h文件中 |
|  | **c)** | 一对文件（Person.h Person.m）只能定义一个类 |
|  | **d)** | 类比较抽象，对象比较具体 |
| 标准答案 | | abd |
| 考核知识点 | | 类的接口，类和对象 |
| 难度 | | 2 |
| 所属章节 | | 0202 |
| 认知层次 | | 3 |

24

|  |  |  |
| --- | --- | --- |
| **二)** | **在OC中，变量的可见度包括（）。（选三项）** | |
|  | **a)** | @public |
|  | **b)** | @protected |
|  | **c)** | @pravite |
|  | **d)** | @optional |
| 标准答案 | | abc |
| 考核知识点 | | 类的定义 |
| 难度 | | 1 |
| 所属章节 | | 0203 |
| 认知层次 | | 1 |

25

|  |  |  |
| --- | --- | --- |
| **二)** | **用于修饰Property 众多Attribute中，其中有一项不正确（）。（选一项）** | |
|  | **a)** | getter;setter |
|  | **b)** | nonatomic;atomic |
|  | **c)** | readwrite;readonly |
|  | **d)** | assign;retain;copy |
| 标准答案 | | b |
| 考核知识点 | | 属性的Attribute |
| 难度 | | 2 |
| 所属章节 | | 0205 |
| 认知层次 | | 2 |

26

|  |  |  |
| --- | --- | --- |
| **二)** | **-(id)initWithName:(NSString \*)name age:(int)age的方法名是（）。（选一项）** | |
|  | **a)** | -initWithName:(NSString \*)name age:(int)age |
|  | **b)** | -initWithName: age: |
|  | **c)** | initWithName: age: |
|  | **d)** | initWithName:age: |
| 标准答案 | | d |
| 考核知识点 | | 方法名 |
| 难度 | | 1 |
| 所属章节 | | 0203 |
| 认知层次 | | 2 |

27

|  |  |  |
| --- | --- | --- |
| **二)** | **关于单例模式的说法正确的有（）。（选三项）** | |
|  | **a)** | 创建单例对象的方法属于类方法 |
|  | **b)** | 单例方法只是为了得到唯一的实例，方法中通常不带参数 |
|  | **c)** | 单例模式是IOS的一种设计模式 |
|  | **d)** | 单例方法中的唯一实例需要autorelease进行释放 |
| 标准答案 | | abc |
| 考核知识点 | | 单例 |
| 难度 | | 2 |
| 所属章节 | | 0208 |
| 认知层次 | | 4 |

28

|  |  |  |
| --- | --- | --- |
| **二)** | **关于内存管理的描述错误的是（）。（选一项）** | |
|  | **a)** | alloc与dealloc是对应的，前者分配内存，后者回收内存 |
|  | **b)** | retain与release是对应的，前者retainCount加1，后者retainCount减1 |
|  | **c)** | 当对象的retainCount为0时，dealloc会被系统调用 |
|  | **d)** | IOS5之后的ARC指的是系统垃圾回收 |
| 标准答案 | | d |
| 考核知识点 | | 内存管理 |
| 难度 | | 3 |
| 所属章节 | | 0303 |
| 认知层次 | | 4 |

29

|  |  |  |
| --- | --- | --- |
| **二)** | **有如下一个类定义**  **@interface MyClass:NSObject**  **@property(nonatomic,retain)NSString \*str;**  **@end**  **@implementation MyClass**  **(1)**  **-(NSString \*)str**  **{**  **return str;**  **}**  **-(void)dealloc**  **{**  **self.str = nil;**  **[super dealloc];**  **}**  **@end** | |
| **在下列选项中，(1)处为str的设置器方法，下列选项中能设置器正确的是（）。（选两项）** | |
|  | **a)** | -(void)setStr:(NSString \*)newStr  {  str = newStr;  } |
|  | **b)** | -(void)setStr:(NSString \*)newStr  {  [str release];  str = [newStr retain];  } |
|  | **c)** | -(void)setStr:(NSString \*)newStr  {  if(str != newStr)  {  [str release];  str = [newStr retain];  }  } |
|  | **d)** | -(void)setStr:(NSString \*)newStr  {  [newStr retain]  [str release];  str = newStr;  } |
| 标准答案 | | cd |
| 考核知识点 | | 设置器 |
| 难度 | | 3 |
| 所属章节 | | 0304 |
| 认知层次 | | 4 |

30

|  |  |  |
| --- | --- | --- |
| **二)** | **NSAutoreleasePool \*pool = [[NSAutoreleasePool alloc] init];**  **NSString \*str = [NSString stringWithFormat:@”abc”];**  **NSString \*str2 = [str retain];**  **[pool release];**  **NSLog(@”str retainCount = %d,str2 retainCount = %d”,[str retainCount],[str2 retainCount]);** | |
| **上述代码打印结果正确的是（）。（选一项）** | |
|  | **a)** | str retainCount = 2,str2 retainCount = 2 |
|  | **b)** | str retainCount = 0,str2 retainCount = 1 |
|  | **c)** | str retainCount = 1,str2 retainCount = 1 |
|  | **d)** | str retainCount = 2,str2 retainCount = 0 |
| 标准答案 | | c |
| 考核知识点 | | 内存管理 |
| 难度 | | 3 |
| 所属章节 | | 0305 |
| 认知层次 | | 4 |

31

|  |  |  |
| --- | --- | --- |
| **二)** | **NSString \*name = [[NSString alloc]initWithString:@”张三”];**  **NSLog(@”%d”,[name retainCount]);** | |
| **上述代码打印结果是（）。（选一项）** | |
|  | **a)** | -1 |
|  | **b)** | 0 |
|  | **c)** | 1 |
|  | **d)** | 2 |
| 标准答案 | | a |
| 考核知识点 | | 字符串 |
| 难度 | | 1 |
| 所属章节 | | 0701 |
| 认知层次 | | 3 |

32

|  |  |  |
| --- | --- | --- |
| **二)** | **NSDictionary \*dict = [NSDictionary dictionaryWithObject:@”a value” forKey:@”aKey”];**  **NSLog(@”%@”,[dict objectForKey:@”aKey”]);**  **[dict release];** | |
| **控制台中最后打印的信息会是（）。（选一项）** | |
|  | **a)** | a value |
|  | **b)** | aKey : a value |
|  | **c)** | 崩溃信息 |
|  | **d)** | a value : aKey |
| 标准答案 | | c |
| 考核知识点 | | 内存管理 |
| 难度 | | 1 |
| 所属章节 | | 0303 |
| 认知层次 | | 3 |

33

|  |  |  |
| --- | --- | --- |
| **二)** | **NSArray \*array = [NSArray arrayWithObjects:@”one”, @”two”, @”three”,nil];**  **NSLog(@”%@”,[array objectAtIndex:1]);**  **NSLog(@”%@”,[array objectAtIndex:3]);** | |
| **控制台中最后打印的信息会是（）。（选一项）** | |
|  | **a)** | one |
|  | **b)** | two |
|  | **c)** | three |
|  | **d)** | 崩溃信息 |
| 标准答案 | | d |
| 考核知识点 | | 数组 |
| 难度 | | 1 |
| 所属章节 | | 0703 |
| 认知层次 | | 2 |

34

|  |  |  |
| --- | --- | --- |
| **二)** | **NSArray \*array = [NSArray arrayWithObjects:@”A”, @”B”, @”C”,@”D”,@”A”,@”B”nil];**  **NSMutableSet \*set = [[NSMutableSet alloc]init];**  **for(NSString \*str in array)**  **{**  **[set addObject:str];**  **}**  **NSLog(@”%d”,[set count]);**  **[set release];** | |
| **控制台打印的结果是（）。（选一项）** | |
|  | **a)** | 4 |
|  | **b)** | 5 |
|  | **c)** | 6 |
|  | **d)** | 7 |
| 标准答案 | | a |
| 考核知识点 | | 集合 |
| 难度 | | 1 |
| 所属章节 | | 0705 |
| 认知层次 | | 2 |

35

|  |  |  |
| --- | --- | --- |
| **二)** | **将@”2B青年”替换成@”文艺青年”的代码是（）。（选两项）** | |
|  | **a)** | [@”2B青年” stringByReplacingOccurrencesOfString:@”2B” withString:@”文艺”]; |
|  | **b)** | NSRange range = {0,2};  [@”2B青年” stringByReplacingCharactersInRange:range withString:@”文艺”]; |
|  | **c)** | NSRange range = {1,2};  [@”2B青年” stringByReplacingCharactersInRange:range withString:@”文艺”]; |
|  | **d)** | [@”2B青年” stringByReplacingOccurrencesOfString:@”文艺” withString:@”2B”]; |
| 标准答案 | | ab |
| 考核知识点 | | 字符串 |
| 难度 | | 3 |
| 所属章节 | | 0701 |
| 认知层次 | | 4 |

36

|  |  |  |
| --- | --- | --- |
| **二)** | **关于通知的描述正确的是（）。（选三项）** | |
|  | **a)** | 通知用于MVC中V与C通信 |
|  | **b)** | name用来唯一标识一个通知对象 |
|  | **c)** | 通知中心以广播形式发送通知 |
|  | **d)** | 通常不手动创建通知，而是使用通知中心创建通知 |
| 标准答案 | | bcd |
| 考核知识点 | | 通知 |
| 难度 | | 2 |
| 所属章节 | | 0502 |
| 认知层次 | | 4 |

37

|  |  |  |
| --- | --- | --- |
| **二)** | **关于KVO的描述正确的是（）。（选两项）** | |
|  | **a)** | KVO用于MVC中V与C通信 |
|  | **b)** | KVO是基于KVC实现的 |
|  | **c)** | KVO本质上采用的是广播形式传递信息 |
|  | **d)** | KVO是key-value-observing的缩写 |
| 标准答案 | | bd |
| 考核知识点 | | KVO |
| 难度 | | 2 |
| 所属章节 | | 0402 |
| 认知层次 | | 4 |

38

|  |  |  |
| --- | --- | --- |
| **二)** | **关于类目、延展的描述错误的是（）。（选一项）** | |
|  | **a)** | 延展主要为类提供“私有”方法 |
|  | **b)** | 类目、延展不但能为类添加方法，而且能为类添加属性 |
|  | **c)** | 无论能否拿到类的源码，类目都可以对其扩展 |
|  | **d)** | 类目和延展声明语法不同 |
| 标准答案 | | d |
| 考核知识点 | | 类目、延展 |
| 难度 | | 2 |
| 所属章节 | | 0601 |
| 认知层次 | | 3 |

39

|  |  |  |
| --- | --- | --- |
| **二)** | **协议与委托的描述正确的是（）。（选三项）** | |
|  | **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 |
| 标准答案 | | abd |
| 考核知识点 | | 协议 |
| 难度 | | 2 |
| 所属章节 | | 0603 |
| 认知层次 | | 3 |

40

|  |  |  |
| --- | --- | --- |
| **二)** | **OC中的委托最好声明成（）。（选一项）** | |
|  | **a)** | retain |
|  | **b)** | assign |
|  | **c)** | copy |
|  | **d)** | readonly |
| 标准答案 | | b |
| 考核知识点 | | 委托 |
| 难度 | | 1 |
| 所属章节 | | 0603 |
| 认知层次 | | 2 |

41

|  |  |  |
| --- | --- | --- |
| 二**)** | 在**UI**基础控件中（）方法，定义是否可与用户交互 。（选一项） | |
|  | **a)** | userInteractionEnabled： |
|  | **b)** | highlighted： |
|  | **c)** | shadowOffset： |
|  | **d)** | textRectForBounds： |
| 标准答案 | | a |
| 考核知识点 | | UI基础框架交互设置 |
| 难度 | | 2 |
| 所属章节 | | 1201 |
| 认知层次 | | 2 |

42

|  |  |  |
| --- | --- | --- |
| 二**)** | 当传输信息量多的时候我们使用**()**请求方式传输。（选一项） | |
|  | **a)** | ftp |
|  | **b)** | get |
|  | **c)** | post |
|  | **d)** | tcp |
| 标准答案 | | c |
| 考核知识点 | | post与get传输方式的区别 |
| 难度 | | 1 |
| 所属章节 | | 0303 |
| 认知层次 | | 1 |

43

|  |  |  |
| --- | --- | --- |
| 二**)** | 实例变量默认的访问修饰符是（）。（选一项） | |
|  | **a)** | @public |
|  | **b)** | @private |
|  | **c)** | @protected |
|  | **d)** | @package |
| 标准答案 | | c |
| 考核知识点 | | 实例变量的访问修饰符 |
| 难度 | | 2 |
| 所属章节 | | 0203 |
| 认知层次 | | 1 |

44

|  |  |  |
| --- | --- | --- |
| 二**)** | 面向对象的特性包括（）（选三项）。 | |
|  | **a)** | 封装 |
|  | **b)** | 继承 |
|  | **c)** | 多态 |
|  | **d)** | 构造 |
| 标准答案 | | abc |
| 考核知识点 | | 面向对象的特性 |
| 难度 | | 1 |
| 所属章节 | | 0201 |
| 认知层次 | | 1 |

45

|  |  |  |
| --- | --- | --- |
| 二**)** | 写出下面程序段的输出结果（）。（选一项）  NSString \* aString = [[NSString alloc] initWithString:@"123"];  [aString autorelease];  NSMutableArray \* arr = [NSMutableArray array];  [arr addObject:aString];  NSLog(@"string = %lu",[aString retainCount]); | |
|  | **a)** | 1 |
|  | **b)** | 0 |
|  | **c)** | **整型最大值** |
|  | **d)** | 2 |
| 标准答案 | | c |
| 考核知识点 | | 常量字符串的引用记数 |
| 难度 | | 3 |
| 所属章节 | | 0301 |
| 认知层次 | | 3 |

46

|  |  |  |
| --- | --- | --- |
| 二**)** | 必须实现的协议方法使用（）关键字声明**.** （选一项） | |
|  | **a)** | @property |
|  | **b)** | @interface |
|  | **c)** | @required |
|  | **d)** | @protocol |
| 标准答案 | | c |
| 考核知识点 | | 协议方法声明 |
| 难度 | | 1 |
| 所属章节 | | 0603 |
| 认知层次 | | 2 |

47

|  |  |  |
| --- | --- | --- |
| 二**)** | 读代码  NSString \* str =[[NSString  alloc]initWithFormat:@"%@",@"ibokan123ios"];  NSString \* str1=[str substringToIndex:3];  NSString \* str2=[str substringWithRange:NSMakeRange(6, 3)];  NSString \* newStr=[str1 stringByAppendingString:str2];  NSLog(@"newStr=%@",newStr);  则newStr的值为()。（选一项） | |
|  | **a)** | ibokan |
|  | **b)** | ib123 |
|  | **c)** | iboios |
|  | **d)** | ibo123 |
| 标准答案 | | d |
| 考核知识点 | | 字符串类的综合使用 |
| 难度 | | 3 |
| 所属章节 | | 0701 |
| 认知层次 | | 4 |

48

|  |  |  |
| --- | --- | --- |
| 二**)** | 读代码：  NSString \*str1=@"hello";  NSString \*str2=[NSString stringWithString:@"hello2"];  NSLog(@"%ld",[str1 compare:str2]);  输出的内容是（）.（选一项） | |
|  | **a)** | 1 |
|  | **b)** | NO |
|  | **c)** | 0 |
|  | **d)** | -1 |
| 标准答案 | | d |
| 考核知识点 | | 字符串的比较 |
| 难度 | | 2 |
| 所属章节 | | 0701 |
| 认知层次 | | 2 |

49

|  |  |  |
| --- | --- | --- |
| 二**)** | mvc设计模式中处于核心位置的组件是（）.（选一项） | |
|  | **a)** | controller |
|  | **b)** | model |
|  | **c)** | view |
|  | **d)** | 全部上面3项 |
| 标准答案 | | a |
| 考核知识点 | | mvc设计模式的理解 |
| 难度 | | 2 |
| 所属章节 | | 1102 |
| 认知层次 | | 2 |

50

|  |  |  |
| --- | --- | --- |
| 二**)** | NSTimer定时器开始的方法是（）.（选一项） | |
|  | **a)** | start |
|  | **b)** | begin |
|  | **c)** | fire |
|  | **d)** | time |
| 标准答案 | | c |
| 考核知识点 | | 定时器的方法 |
| 难度 | | 1 |
| 所属章节 | | 0804 |
| 认知层次 | | 1 |

51

|  |  |  |
| --- | --- | --- |
| 二**)** | 判断对象是否遵循协议的方法是（）.（选一项） | |
|  | **a)** | conformsToProtocol |
|  | **b)** | agreeToProtocol |
|  | **c)** | underToProtocol |
|  | **d)** | beginToProtocol |
| 标准答案 | | a |
| 考核知识点 | | 判断对象是否遵循协议的方法 |
| 难度 | | 1 |
| 所属章节 | | 0603 |
| 认知层次 | | 1 |

52

|  |  |  |
| --- | --- | --- |
| 二**)** | 读代码  Student \*stu=[[Student alloc]init];  [stu retain];  [stu autorelease];  NSLog(@"%lu",stu.retainCount);  输出的结果是（）.（选一项） | |
|  | **a)** | 1 |
|  | **b)** | 2 |
|  | **c)** | 3 |
|  | **d)** | 4 |
| 标准答案 | | b |
| 考核知识点 | | 内存管理关键字的使用 |
| 难度 | | 2 |
| 所属章节 | | 0302 |
| 认知层次 | | 3 |

53

|  |  |  |
| --- | --- | --- |
| 二**)** | 通知包含的三个属性是（）.（选三项） | |
|  | **a)** | name |
|  | **b)** | object |
|  | **c)** | dictionary |
|  | **d)** | set |
| 标准答案 | | abc |
| 考核知识点 | | 通知包含的三个属性 |
| 难度 | | 1 |
| 所属章节 | | 0501 |
| 认知层次 | | 2 |

54

|  |  |  |
| --- | --- | --- |
| 二) | 类目的作用包括下面那项（）.（选三项） | |
|  | a) | 通过类目的作用,可以将类的实现分散到不同的文件里 |
|  | b) | 创建对私有方法的前向引用 |
|  | c) | 向对象添加非正式协议 |
|  | d) | 实现视图，控制器,模型的分离 |
| 标准答案 | | abc |
| 考核知识点 | | 类目的作用 |
| 难度 | | 2 |
| 所属章节 | | 0601 |
| 认知层次 | | 3 |

55

|  |  |  |
| --- | --- | --- |
| 二**)** | bundle是（）.（选一项） | |
|  | **a)** | 目录 |
|  | **b)** | 文件 |
|  | **c)** | 视频 |
|  | **d)** | 图片 |
| 标准答案 | | a |
| 考核知识点 | | bundle的说明 |
| 难度 | | 1 |
| 所属章节 | | 0903 |
| 认知层次 | | 2 |

56

|  |  |  |
| --- | --- | --- |
| 二) | NSMutableDictionary字典中删除对象可以使用下面那个方法（）.（选一项） | |
|  | a) | filteredArrayUsingPredicate: |
|  | b) | arrayByAddingObject: |
|  | c) | initWithCapacity: |
|  | d) | setValue:forKey: |
| 标准答案 | | d |
| 考核知识点 | | 字典函数的用法 |
| 难度 | | 3 |
| 所属章节 | | 0703 |
| 认知层次 | | 4 |

57

|  |  |  |
| --- | --- | --- |
| 二**)** | 读代码  Student \*stu=[[Student alloc]init];  [stu retain];  [stu autorelease];  NSMutableSet \*mt=[NSMutableSet setWithObjects:stu, nil];  NSLog(@"%lu",stu.retainCount);  输出的结果是（）.（选一项） | |
|  | **a)** | 1 |
|  | **b)** | 2 |
|  | **c)** | 3 |
|  | **d)** | 4 |
| 标准答案 | | c |
| 考核知识点 | | 数组内存管理 |
| 难度 | | 2 |
| 所属章节 | | 0706 |
| 认知层次 | | 3 |

58

|  |  |  |
| --- | --- | --- |
| 二**)** | 读代码  NSString \*str=@"hello";  NSMutableSet \*mt=[NSMutableSet setWithObjects:str, nil];  [mt addObject:str];  NSLog(@"%lu",mt.count);  结果是（）（选一项）。 | |
|  | **a)** | 1 |
|  | **b)** | 2 |
|  | **c)** | 3 |
|  | **d)** | 4 |
| 标准答案 | | a |
| 考核知识点 | | set的特性 |
| 难度 | | 2 |
| 所属章节 | | 0705 |
| 认知层次 | | 3 |

59

|  |  |  |
| --- | --- | --- |
| 二**)** | 下列有效的键路径是（）（选一项）。 | |
|  | **a)** | [foo valueForKeyPath:@”1SomeMember”] |
|  | **b)** | [foo valueForKeyPath:@”someMember.someAttribute”] |
|  | **c)** | [foo valueForKeyPath:@”SOMEMEMBER@someAttibute”] |
|  | **d)** | [foo valueForKeyPath:@”some Member”] |
| 标准答案 | | b |
| 考核知识点 | | 有效的键路径 |
| 难度 | | 2 |
| 所属章节 | | 0401 |
| 认知层次 | | 2 |

60

|  |  |  |
| --- | --- | --- |
| 二**)** | 使用通知的三个步骤不包括（）（选一项）。 | |
|  | **a)** | 注册通知 |
|  | **b)** | 发送通知 |
|  | **c)** | 移除通知 |
|  | **d)** | 开始通知 |
| 标准答案 | | d |
| 考核知识点 | | 使用通知的三个步骤 |
| 难度 | | 1 |
| 所属章节 | | 0503 |
| 认知层次 | | 2 |

61

|  |  |  |
| --- | --- | --- |
| **二)** | **使用 Objective-C 语言输出”This is a Objective-C program.”正确的是：** | |
|  | **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)** | NSLog(@"This is a Objective-C program."); |
| 标准答案 | | d |
| 考核知识点 | | OC基础。 |
| 难度 | | 1 |
| 所属章节 | | 0201 |
| 认知层次 | | 1 |

#### 62

|  |  |  |
| --- | --- | --- |
| **二)** | **关于类和对象的概念，下列属于对象的是：** | |
|  | **a)** | 奥迪A6 |
|  | **b)** | 保龄球 |
|  | **c)** | 世界冠军刘翔 |
|  | **d)** | 西红柿 |
| 标准答案 | | c |
| 考核知识点 | | 类和对象的概念 |
| 难度 | | 1 |
| 所属章节 | | 0202 |
| 认知层次 | | 1 |

63

|  |  |  |
| --- | --- | --- |
| **二)** | **Objective-C 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？** | |
|  | **a)** | @public |
|  | **b)** | @private |
|  | **c)** | @protected |
|  | **d)** | @friendly |
| 标准答案 | | c |
| 考核知识点 | | OC类的实例变量。 |
| 难度 | | 2 |
| 所属章节 | | 0202 |
| 认知层次 | | 1 |

64

|  |  |  |
| --- | --- | --- |
| **二)** | **下列哪个选项不能作为property的可选属性** | |
|  | **a)** | alloc |
|  | **b)** | retain |
|  | **c)** | atomic |
|  | **d)** | strong |
| 标准答案 | | a |
| 考核知识点 | | OC的属性。 |
| 难度 | | 2 |
| 所属章节 | | 0202 |
| 认知层次 | | 1 |

65

|  |  |  |
| --- | --- | --- |
| **二)** | **下列哪个选项与@synthesize配合使用** | |
|  | **a)** | @end |
|  | **b)** | @interface |
|  | **c)** | @implementation |
|  | **d)** | @property |
| 标准答案 | | d |
| 考核知识点 | | OC类的属性。 |
| 难度 | | 1 |
| 所属章节 | | 0202 |
| 认知层次 | | 1 |

66

|  |  |  |
| --- | --- | --- |
| **二)** | **经过下列哪种操作后需要使用release方法。** | |
|  | **a)** | delegate |
|  | **b)** | assign |
|  | **c)** | retain |
|  | **d)** | dealloc |
| 标准答案 | | c |
| 考核知识点 | | OC内存管理。 |
| 难度 | | 1 |
| 所属章节 | | 0203 |
| 认知层次 | | 1 |

67

|  |  |  |
| --- | --- | --- |
| **二)** | **类何时调用dealloc方法？** | |
|  | **a)** | [property release]后 |
|  | **b)** | [instance release]后 |
|  | **c)** | [super dealloc]时 |
|  | **d)** | 实例对象retaincount=0时. |
| 标准答案 | | d |
| 考核知识点 | | OC内存管理。 |
| 难度 | | 2 |
| 所属章节 | | 0203 |
| 认知层次 | | 2 |

68

|  |  |  |
| --- | --- | --- |
| **二)** | **下列哪个选项不需要release方法** | |
|  | **a)** | [[NSString alloc]init]; |
|  | **b)** | [[NSString alloc] initWithString:@”hehe”]; |
|  | **c)** | [[NSArray arrayWithArray: arr1]; |
|  | **d)** | [[NSMutableDictionary alloc] init]; |
| 标准答案 | | c |
| 考核知识点 | | OC内存管理。 |
| 难度 | | 1 |
| 所属章节 | | 0203 |
| 认知层次 | | 2 |

69

|  |  |  |
| --- | --- | --- |
| **二)** | **下列哪个操作不会引起引用计数（retaincount）的变化** | |
|  | **a)** | NSString \*str1 = [[NSString stringWithFormat:@"hehe"]; |
|  | **b)** | int a=1; |
|  | **c)** | NSMutableData \*data1 = [NSMutableData allocWithZone:data2]; |
|  | **d)** | NSArray \*arr1 = [[NSArray alloc]initWithArray:arr2]; |
| 标准答案 | | b |
| 考核知识点 | | OC类的属性。 |
| 难度 | | 3 |
| 所属章节 | | 0203 |
| 认知层次 | | 3 |

70

|  |  |  |
| --- | --- | --- |
| **二)** | **如果使用[str autorelease]语句，str在何时释放内存？** | |
|  | **a)** | 该语句执行后立即释放 |
|  | **b)** | 该语句所在的方法结束后释放 |
|  | **c)** | autoreleasepool释放时 |
|  | **d)** | str.retaincount的值为0时 |
| 标准答案 | | d |
| 考核知识点 | | OC内存管理 |
| 难度 | | 3 |
| 所属章节 | | 0203 |
| 认知层次 | | 3 |

71

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSString \* aString = [[NSString alloc] initWithString:@"hehe"]; [aString autorelease]; NSMutableArray \* arr = [NSMutableArray array]; [arr addObject:aString];**  **NSLog(@"string = %d",[aString retainCount]);** | |
|  | **a)** | string = -1 |
|  | **b)** | string = 0 |
|  | **c)** | string = 1 |
|  | **d)** | string = 2 |
| 标准答案 | | a |
| 考核知识点 | | OC内存管理。 |
| 难度 | | 3 |
| 所属章节 | | 0203 |
| 认知层次 | | 3 |

72

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSString \* aString = [[NSString alloc] initWithFormat:@"123"];**  **NSLog(@"%d",aString.retainCount);**  **aString = @"456";**  **NSLog(@"%d",aString.retainCount);**  **[aString release];**  **aString = @"789";**  **NSLog(@"%d",aString.retainCount);** | |
|  | **a)** | 1,2,1 |
|  | **b)** | 1,1,0 |
|  | **c)** | 1,-1,0 |
|  | **d)** | 1,-1,-1 |
| 标准答案 | | d |
| 考核知识点 | | OC内存管理。 |
| 难度 | | 3 |
| 所属章节 | | 0203 |
| 认知层次 | | 3 |

73

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSArray \* arr1 = [[NSArray alloc] initWithObjects:@"1",@"2",@"3", nil];**  **NSString \*str;**  **str = [arr1 objectAtIndex:2];**  **NSLog(@"%@",str);** | |
|  | **a)** | 1 |
|  | **b)** | 2 |
|  | **c)** | 3 |
|  | **d)** | 程序崩溃 |
| 标准答案 | | c |
| 考核知识点 | | OC 数组方法及下标。 |
| 难度 | | 2 |
| 所属章节 | | 0207 |
| 认知层次 | | 2 |

74

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableArray \* arr1 = [[NSMutableArray alloc] initWithObjects:@"1",@"2",@"3", nil];**  **[arr1 addObject:@"0"];**  **[arr1 replaceObjectAtIndex:2 withObject:@"3"];**  **NSLog(@"%@",arr1);** | |
|  | **a)** | 0,1,2,3 |
|  | **b)** | 1,2,3,0 |
|  | **c)** | 1,3,3,0 |
|  | **d)** | 0,3,2,3 |
| 标准答案 | | b |
| 考核知识点 | | OC数组方法。 |
| 难度 | | 2 |
| 所属章节 | | 0207 |
| 认知层次 | | 3 |

75

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableDictionary \* dic = [[NSMutableDictionary alloc] initWithObjectsAndKeys:@"1",@"2",@"2",@"3",@"3",@"4", nil];**  **[dic removeObjectForKey:@"3"];**  **NSLog(@"%@",dic);** | |
|  | **a)** | 1=2 3=4 |
|  | **b)** | 1=2 2=3 |
|  | **c)** | 2=1 4=3 |
|  | **d)** | 2=1 3=2 |
| 标准答案 | | c |
| 考核知识点 | | OC 字典的方法应用，键值的辨别。 |
| 难度 | | 2 |
| 所属章节 | | 0207 |
| 认知层次 | | 3 |

76

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableDictionary \* dic = [[NSMutableDictionary alloc] initWithObjectsAndKeys:@"1",@"2",@"2",@"3",@"3",@"4", nil];**  **[dic setObject:@"1" forKey:@"3"];**  **[dic removeObjectForKey:@"1"];**  **NSLog(@"%@",dic);** | |
|  | **a)** | 1=3 2=3 3=4 |
|  | **b)** | 1=2 1=3 3=4 |
|  | **c)** | 2=1 3=1 4=3 |
|  | **d)** | 3=1 3=2 4=3 |
| 标准答案 | | c |
| 考核知识点 | | OC 字典的方法应用，键值的辨别。 |
| 难度 | | 3 |
| 所属章节 | | 0207 |
| 认知层次 | | 4 |

77

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **int a = 10;**  **NSNumber \*b = [[NSNumber alloc] initWithInt:a];**  **NSLog(@"%d", a+++b);** | |
|  | **a)** | 21 |
|  | **b)** | 20 |
|  | **c)** | 30 |
|  | **d)** | 编译错误 |
| 标准答案 | | d |
| 考核知识点 | | OC NSNumber类指针操作。 |
| 难度 | | 3 |
| 所属章节 | | 0207 |
| 认知层次 | | 2 |

78

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableDictionary \* dic = [[NSMutableDictionary alloc] initWithObjectsAndKeys:@"1",@"2",@"2",@"3",@"3",@"4", nil];**  **NSString \*str = @"3";**  **int a = 3;**  **[dic setObject:a forKey:str];**  **NSLog(@"%@",dic);** | |
|  | **a)** | 2=1 3=3 4=3 |
|  | **b)** | 1=2 3=3 3=4 |
|  | **c)** | 1=2 3=3 3=3 |
|  | **d)** | 程序崩溃 |
| 标准答案 | | d |
| 考核知识点 | | OC 字典应用，id类型。 |
| 难度 | | 3 |
| 所属章节 | | 0207 |
| 认知层次 | | 4 |

79

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableDictionary \* dic = [[NSMutableDictionary alloc] initWithObjectsAndKeys:@"1",@"2",@"2",@"3",@"3",@"4", nil];**  **NSArray \*arr = [[NSArray alloc] initWithArray:[dic allKeys]];**  **NSString \*str = [[NSString alloc] initWithFormat:[arr objectAtIndex:0],[arr objectAtIndex:1],[arr objectAtIndex:2]];**  **NSLog(@"%@",str);** | |
|  | **a)** | 123 |
|  | **b)** | 234 |
|  | **c)** | 2 |
|  | **d)** | 3 |
| 标准答案 | | c |
| 考核知识点 | | OC字符串，数组，字典方法的综合应用。 |
| 难度 | | 3 |
| 所属章节 | | 0207 |
| 认知层次 | | 4 |

80

|  |  |  |
| --- | --- | --- |
| **二)** | **下面程序段的输出结果是什么？**  **NSMutableDictionary \* dic = [[NSMutableDictionary alloc] initWithObjectsAndKeys:@"1",@"2",@"1",@"3",@"1",@"4", nil];**  **NSArray \*arr1 = [[NSArray alloc] initWithArray:[dic allKeys]];**  **NSArray \*arr2 = [[NSArray alloc] initWithArray:[dic allValues]];**  **NSMutableDictionary \*dic2 = [[NSMutableDictionary alloc] initWithObjects:arr1 forKeys:arr2];**  **NSLog(@"%@",dic2);** | |
|  | **a)** | 1=2 1=3 1=4 |
|  | **b)** | 2=1 3=1 4=1 |
|  | **c)** | 1=4 |
|  | **d)** | 4=1 |
| 标准答案 | | c |
| 考核知识点 | | OC 字典、数组方法的综合应用。 |
| 难度 | | 3 |
| 所属章节 | | 0207 |
| 认知层次 | | 4 |

81

|  |  |  |
| --- | --- | --- |
| 二**)** | @interface Person : NSObject  {  NSString \* name;  }  @end  建一个**Person**类，如何从类的外部访问实例变量**name**（）（选两项）。 | |
|  | **a)** | 写一个name属性 |
|  | **b)** | 为实例变量name添加关键字@private |
|  | **c)** | 为实例变量name添加setter和getter方法 |
|  | **d)** | 无法访问 |
| 标准答案 | | ac |
| 考核知识点 | | 属性的使用，实例变量关键字，setter和getter方法 |
| 难度 | | 2 |
| 所属章节 | | 0203、0205 |
| 认知层次 | | 1 |

82

|  |  |  |
| --- | --- | --- |
| 二**)** | 下面哪个便利构造器的实现是正确的（）（选一项）。 | |
|  | **a)** | +(Person \*)personWithName:(NSString \*)theName{  self = [[Person alloc] init];  self.name = theName;  return self;  } |
|  | **b)** | +(Person \*)personWithName:(NSString \*)theName{  Person \* per = [[Person alloc] init];  per.name = theName;  return self ;  } |
|  | **c)** | +(Person \*)personWithName:(NSString \*)theName{  Person \* per = [[Person alloc] init];  per.name = theName;  return [per autorelease] ;  } |
|  | **d)** | +(Person \*)personWithName:(NSString \*)theName{  Person \* per = [[Person alloc] init];  per.name = theName;  return [self autorelease];  } |
| 标准答案 | | c |
| 考核知识点 | | 便利构造器 |
| 难度 | | 2 |
| 所属章节 | | 0207 |
| 认知层次 | | 2 |

83

|  |  |  |
| --- | --- | --- |
| 二**)** | Person \* per = [[Person alloc] init];  per.name = [[[NSString alloc] initWithFormat:@"张三"] autorelease];  NSLog(@"per.name = %d",[per.name retainCount]);  打印结果为（）（选一项）。 | |
|  | **a)** | per.name = 1 |
|  | **b)** | per.name = 2 |
|  | **c)** | per.name =-1 |
|  | **d)** | per.name = 0 |
| 标准答案 | | b |
| 考核知识点 | | 内存管理 |
| 难度 | | 2 |
| 所属章节 | | 0304 |
| 认知层次 | | 3 |

84

|  |  |  |
| --- | --- | --- |
| 二**)** | **有这样2个属性**  @property(nonatomic,strong)NSString \* str1;  @property(nonatomic,weak)NSString \* str2;  **在本类有如下代码**  self.str1 = [[NSString alloc] initWithFormat:@"str1"];  self.str2 = self.str1;  self.str1 = nil;  NSLog(@"self.str2 = %@",self.str2);  **输出结果为（）（选一项）。** | |
|  | **a)** | self.str2 = str1; |
|  | **b)** | self.str2 = (null); |
|  | **c)** | 输出乱码 |
|  | **d)** | 程序崩溃 |
| 标准答案 | | b |
| 考核知识点 | | 强弱引用 |
| 难度 | | 3 |
| 所属章节 | | 0306 |
| 认知层次 | | 4 |

85

|  |  |  |
| --- | --- | --- |
| 二**)** | **下面描述正确的选项是（）（选两项）。** | |
|  | **a)** | dealloc方法由系统自动调用 |
|  | **b)** | 一个类在释放的时候，会调用自己的dealloc方法 |
|  | **c)** | 在dealloc方法里，应该在[super dealloc]后释放本类所拥有的对象 |
|  | **d)** | 开启ARC后，可以重载dealloc，必须在dealloc里写 [super dealloc]方法 |
| 标准答案 | | ab |
| 考核知识点 | | dealloc方法 |
| 难度 | | 2 |
| 所属章节 | | 0306 |
| 认知层次 | | 2 |

86

|  |  |  |
| --- | --- | --- |
| 二**)** | **下面哪个选项是正确的（）（选两项）。** | |
|  | **a)** | 用继承可以扩展一个类，要调用扩展的方法，既可以用父类，也可以用子类 |
|  | **b)** | 用类目可以扩展一个类，可以直接用该类，去调用扩展的方法 |
|  | **c)** | 延展是类目的特殊形式 |
|  | **d)** | 可以在一个类的外部，调用这个类延展的方法 |
| 标准答案 | | bc |
| 考核知识点 | | 继承、类目、延展 |
| 难度 | | 2 |
| 所属章节 | | 0601、0602 |
| 认知层次 | | 2 |

87

|  |  |  |
| --- | --- | --- |
| 二**)** | @implementation Person  @synthesizeage;  -(void)setAge:(int)newAge{  self.age = newAge;  }  @end  **运行结果是（）（选一项）。** | |
|  | **a)** | 程序陷入死循环 |
|  | **b)** | 程序正常运行 |
|  | **c)** | 程序崩溃 |
|  | **d)** | 不确定 |
| 标准答案 | | a |
| 考核知识点 | | 设置器 |
| 难度 | | 3 |
| 所属章节 | | 0204 |
| 认知层次 | | 4 |

88

|  |  |  |
| --- | --- | --- |
| 二**)** | NSDateFormatter \* formatter = [[NSDateFormatter alloc] init];  [formatter setAMSymbol:@"AM"];  [formatter setPMSymbol:@"PM"];  [formatter setDateFormat:@"YY-MM-dd hh:mm:ss aaa"];  NSString \* currentDate = [formatter stringFromDate:[NSDate date]];  NSLog(@"%@",currentDate);  **打印结果是（）（选一项）。** | |
|  | **a)** | 2012-05-31 17:45:04 PM |
|  | **b)** | 12-05-31 17:45:04 |
|  | **c)** | 2012-05-31 17:45:04 |
|  | **d)** | 12-05-31 05:45:04 PM |
| 标准答案 | | d |
| 考核知识点 | | NSDate |
| 难度 | | 2 |
| 所属章节 | | 0801 |
| 认知层次 | | 2 |

89

|  |  |  |
| --- | --- | --- |
| 二**)** | **哪个选项可以获取沙盒中的Documents文件夹的路径（）（选两项）。** | |
|  | **a)** | NSArray \* paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);  NSString \* documents = [paths objectAtIndex:0]; |
|  | **b)** | NSArray \* paths = NSSearchPathForDirectoriesInDomains(NSDocumentationDirectory, NSUserDomainMask, YES);  NSString \* documents = [paths objectAtIndex:0]; |
|  | **c)** | NSString \*homeDirectory = NSHomeDirectory();  NSString \*fileDirectory = [homeDirectory stringByAppendingPathComponent:@"Documents"]; |
|  | **d)** | NSArray \* paths = NSSearchPathForDirectoriesInDomains(NSUserDirectory, NSUserDomainMask, YES);  NSString \* documents = [paths objectAtIndex:0]; |
| 标准答案 | | ac |
| 考核知识点 | | 沙盒路径 |
| 难度 | | 1 |
| 所属章节 | | 0903 |
| 认知层次 | | 3 |

90

|  |  |  |
| --- | --- | --- |
| 二**)** | **怎样声明私有方法（）（选三项）。** | |
|  | **a)** | 类目 |
|  | **b)** | 协议 |
|  | **c)** | 多态 |
|  | **d)** | 延展 |
| 标准答案 | | d |
| 考核知识点 | | 延展的概念 |
| 难度 | | 1 |
| 所属章节 | | 0602 |
| 认知层次 | | 2 |

91

|  |  |  |
| --- | --- | --- |
| 二**)** | 下面哪些选项是类（）（选**2**项）。 | |
|  | **a)** | 导演 |
|  | **b)** | 张艺谋 |
|  | **c)** | 擎天柱 |
|  | **d)** | 变形金刚 |
| 标准答案 | | ad |
| 考核知识点 | | 类的定义 |
| 难度 | | 1 |
| 所属章节 | | 0202 |
| 认知层次 | | 2 |

92

|  |  |  |
| --- | --- | --- |
| 二**)** | 下面哪些选项是属性的正确声明方式（）（选**2**项）。 | |
|  | **a)** | @property(nonatomic,retain)NSString \* myString; |
|  | **b)** | @property(nonatomic,assign)NSString \* myString; |
|  | **c)** | @property(nonatomic,assign)int mynumber; |
|  | **d)** | @property(nonatomic,retain)int mynumber; |
| 标准答案 | | ac |
| 考核知识点 | | 属性的使用 |
| 难度 | | 1 |
| 所属章节 | | 0205 |
| 认知层次 | | 2 |

93

|  |  |  |
| --- | --- | --- |
| 二**)** | 自定义一个**Person**类，下面哪个选项是便利构造器的声明方式（）（选**1**项）。 | |
|  | **a)** | -(id)personWithName:(NSString \*)theName; |
|  | **b)** | -(id)personwithName:(NSString \*)theName; |
|  | **c)** | +(id)personwithName:(NSString \*)theName; |
|  | **d)** | +(id)personWithName:(NSString \*)theName; |
| 标准答案 | | d |
| 考核知识点 | | 便利构造器的定义 |
| 难度 | | 2 |
| 所属章节 | | 0207 |
| 认知层次 | | 1 |

94

|  |  |  |
| --- | --- | --- |
| 二**)** | **下面哪个选项无内存问题的（）（选2**项）。 | |
|  | **a)** | -(NSString \*)description{  NSString \* str = [[NSString alloc] initWithFormat:@"description"];  return str;  } |
|  | **b)** | -(NSString \*)description{  NSString \* str = [[NSString alloc] initWithFormat:@"description"];  [str release];  return str;  } |
|  | **c)** | -(NSString \*)description{  NSString \* str = [[NSString alloc] initWithFormat:@"description"];  [str autorelease];  return str;  } |
|  | **d)** | -(NSString \*)description{  NSString \* str = [NSString stringWithFormat:@"description"];  return str;  } |
| 标准答案 | | cd |
| 考核知识点 | | 内存管理 |
| 难度 | | 2 |
| 所属章节 | | 0305 |
| 认知层次 | | 3 |

95

|  |  |  |
| --- | --- | --- |
| 二**)** | @interface Person : NSObject  @property(nonatomic,retain)NSString \* name;  @end  **上面声明一个Person类**  Person \* per = [[Person alloc] init];  per.name = [[NSString alloc] initWithFormat:@"张三"];  NSLog(@"per.name = %d",[per.name retainCount]);  **上面代码应该打印出什么（）（选1**项）。 | |
|  | **a)** | per.name = -1; |
|  | **b)** | per.name = 0; |
|  | **c)** | per.name = 1; |
|  | **d)** | per.name = 2; |
| 标准答案 | | d |
| 考核知识点 | | 内存管理 |
| 难度 | | 2 |
| 所属章节 | | 0304 |
| 认知层次 | | 4 |

96

|  |  |  |
| --- | --- | --- |
| 二**)** | **用哪个关键字定义协议中必须实现的方法（）（选2**项）。 | |
|  | **a)** | @optional |
|  | **b)** | @protected |
|  | **c)** | @required |
|  | **d)** | 不用写关键字 |
| 标准答案 | | cd |
| 考核知识点 | | 协议 |
| 难度 | | 1 |
| 所属章节 | | 0603 |
| 认知层次 | | 2 |

97

|  |  |  |
| --- | --- | --- |
| 二**)** | NSMutableString \* str = [[NSMutableString alloc] initWithFormat:@"123"];  NSLog(@"str = %p",str);  [str appendFormat:@"456"];  NSLog(@"str = %p",str);  **上面代码对字符串str进行拼接后，2次打印的地址是否相同（）（选1**项）。 | |
|  | **a)** | 相同 |
|  | **b)** | 不同 |
|  | **c)** | 随机打印 |
|  | **d)** | 有时相同，有时不同 |
| 标准答案 | | a |
| 考核知识点 | | 可变字符串 |
| 难度 | | 1 |
| 所属章节 | | 0701 |
| 认知层次 | | 2 |

98

|  |  |  |
| --- | --- | --- |
| 二**)** | **下面哪个选项对集合的描述是正确的（）（选1**项）。 | |
|  | **a)** | 集合内不能存不同类型的对象 |
|  | **b)** | 集合内可以存整型的数 |
|  | **c)** | 字典内键值可以相同，但是每个键值对应的对象必须不同 |
|  | **d)** | 集合内只能存对象 |
| 标准答案 | | d |
| 考核知识点 | | 对象 |
| 难度 | | 2 |
| 所属章节 | | 0703、0704、0705 |
| 认知层次 | | 1 |

99

|  |  |  |
| --- | --- | --- |
| 二**)** | **对沙盒下的子目录描述正确的是哪个（）（选2**项）。 | |
|  | **a)** | Documents目录下存放需要持久保存的数据 |
|  | **b)** | AppName.app下的文件直接修改 |
|  | **c)** | tmp目录下存放临时文件 |
|  | **d)** | iTunes在与iPhone同步时，会备份沙盒下所有个子目录中的文件 |
| 标准答案 | | ac |
| 考核知识点 | | 沙盒 |
| 难度 | | 1 |
| 所属章节 | | 0903 |
| 认知层次 | | 1 |

100

|  |  |  |
| --- | --- | --- |
| 二**)** | **下面正确的选项有哪些（）（选3**项）。 | |
|  | **a)** | 当传输数据量较大时，应该选择异步连接 |
|  | **b)** | get请求通常用于请求URL并得到资源 |
|  | **c)** | Post请求用于想服务器添加新内容 |
|  | **d)** | get请求安全性高于post请求 |
| 标准答案 | | abc |
| 考核知识点 | | 网络 |
| 难度 | | 1 |
| 所属章节 | | 1004、1005 |
| 认知层次 | | 1 |