# 程序面试方案

为了提高公司程序员面试的效率，使面试评分更客观更标准，提出如下方案，对程序面试流程进行改进，定义了标准的考察维度、统一的分数标准以及准备相对应的题库，使得每一次的面试更加标准和高效。

**一、面试考察的能力及说明：**

面试题目分为五个维度，考察面试者各方面的编程能力。考察的能力点简介和考察目的如下所示：

1. **C/C++或Java语法和基础知识：**

**考察内容：**基础的编程语法、常见的API使用；

**考察目的：**了解面试者的编程经验和基本能力；

**考察意义：**有编程经验的面试者能够更快的学习C#等其他语言以及相关API的使用；

1. **数据结构和算法：**

**考察内容：**常见的数据结构，以及这些数据结构的设计思路和应用范围；常用的算法， 基本的实现方式、算法的时间复杂度和空间复杂度等；

**考察目的：**考察面试者数据结构和算法相关的基础知识及逻辑思维能力；

**考察意义：**算法和数据结构是编程的内功，掌握常见的数据结构和算法可以更容易解决 开发过程中遇到的问题，同时也说明面试者具备更强的逻辑思维能力；

1. **计算机网络和操作系统：**

**考察内容：**TCP/IP相关的知识，HTTP协议相关的知识，操作系统线程等；

**考察目的：**考察面试者对计算机网络和操作系统底层的了解；

**考察意义：**对计算机底层理解越深，越能够理解上层API设计的意义，成长空间越大；

1. **设计模式和软件架构：**

**考察内容：**软件架构相关的知识；

**考察目的：**考察面试者的工程能力，以及宏观上对代码的理解；

**考察意义：**对该知识点的理解越深，代码编写上越能适应变化，代码越稳定；

1. **项目和游戏开发经验：**

**考察内容：**考察面试者在简历中呈现的项目中参与的内容，对项目的熟悉程度以及一些 常见开发工具的使用；

**考察目的：**面试者是否真实的参与了简历呈现的项目，参与程度如何，自学Unity或者 游戏开发的程度如何；

**考察目的：**项目经验多的更能够适应团队开发的工作，更好的理解需求；如果有自学 Unity或者游戏开发经验，能够更快的完成培训参与到项目中，同时也能够更好的 适应公司的开发流程；

1. **面试评分标准和考题内容：**

考虑到上述罗列的考察面试者的五个维度的能力值并不是同样重要，因此对每个能力值给予不同的总分和考察题量，总计满分100分。具体分数分布和面试题量如下：

1. C/C++或Java语法和基础知识：考察8题，每题5分，总计40分；
2. 数据结构和算法：考察6题，每题6分，共计36分；
3. 计算机网络和操作系统：考察3题，每题4分，共计12分；
4. 设计模式和软件架构：考察2题，每题3分，共计6分；
5. 项目和游戏开发经验：考察2题，每题3分，共计6分；

详细说明：

1. 由于编程语法基础、数据结构和算法是程序开发的核心能力，因此分别占比40%和36%，题目数量也相对较多，难易结合；
2. 计算机网络和操作系统、设计模式和软件架构，考察的更多的是面试者的潜力和计算机基本功，在工作中用到的相对较少，因此分数占比相对较低为12%每项；
3. 项目和游戏开发经验对于面试者来说，随着入职时间的增加，作用在不断减少，根据对进入公司超过半年的新人的观察，大学期间的项目经验影响已经不大，因此分数占比最少为6%；
4. 每道题答对得满分，错误得0分，部分错误由两位面试官的分数平均值决定；
5. 当前设置的题量是根据以往面试的经验考虑的，保证面试时间在10分钟以内能够结束，后期会根据新的面试情况进行反馈调整题目数量和分值占比；

**三、面试的基本流程：**

为了保证面试成绩的准确和标准化，针对每一位面试者的题目内容可能存在重复，因此需要针对每一位面试者单独进行面试，面试流程如下（仅包含程序面试内容流程）：

1. 根据面试者介绍选择C/C++或者Java题库；
2. 轮流考察五个维度的知识点，每一题大约30秒左右，答对、答错或超时考察下一题，连续3题答错直接跳过本能力值考察，进行下一能力查考；
3. 若面试者没有参与项目经验或者游戏开发经验，则不进行第五个能力的面试，直接记为0分；
4. 在上述面试流程中，若总时间达到规定时间，例如10分钟仍然有题目尚未考察的，直接终止面试，余下题目记0分；
5. 五个知识点考察完毕或到达指定时间，计算总得分，然后继续面试下一位面试者；

**四、面试结果成绩单：**

提供了一种可能的面试成绩单样式供参考：

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名：** | **学校：** | **面试日期：** | **总分：** |
| 1. **C/C++或Java语法和基础知识** | | | **阶段分：** |
| 题目1编号： | | 得分： | |
| 题目2编号： | | 得分： | |
| 题目3编号： | | 得分： | |
| 题目4编号： | | 得分： | |
| 题目5编号： | | 得分： | |
| 题目6编号： | | 得分： | |
| 题目7编号： | | 得分： | |
| 题目8编号： | | 得分： | |
| 1. **数据结构和算法** | | | **阶段分：** |
| 题目1编号： | | 得分： | |
| 题目2编号： | | 得分： | |
| 题目3编号： | | 得分： | |
| 题目4编号： | | 得分： | |
| 题目5编号： | | 得分： | |
| 题目6编号： | | 得分： | |
| 1. **计算机网络和操作系统** | | | **阶段分：** |
| 题目1编号： | | 得分： | |
| 题目2编号： | | 得分： | |
| 题目3编号： | | 得分： | |
| 1. **设计模式和软件架构** | | | **阶段分：** |
| 题目1编号： | | 得分： | |
| 题目2编号： | | 得分： | |
| 1. **项目和游戏开发经验** | | | **阶段分：** |
| 题目1编号： | | 得分： | |
| 题目2编号： | | 得分： | |

**相关模块的题库见后面几页。**

**Java语法和基础知识**

1. 介绍String类中replace split substring 的用法
2. for循环中 break和continue的作用
3. int和Integer有什么区别
4. 请你讲讲数组（Array）和列表（ArrayList）的区别？什么时候应该使用Array而不是ArrayList

Array和ArrayList的不同点：  
Array可以包含基本类型和对象类型，ArrayList只能包含对象类型。  
Array大小是固定的，ArrayList的大小是动态变化的。  
ArrayList提供了更多的方法和特性，比如：addAll()，removeAll()，iterator()等等。  
对于基本类型数据，集合使用自动装箱来减少编码工作量。但是，当处理固定大小的基本数据类型的时候，这种方式相对比较慢。

1. 请解释什么是值传递和引用传递

值传递是对基本型变量而言的,传递的是该变量的一个副本,改变副本不影响原变量.  
引用传递一般是对于对象型变量而言的,传递的是该对象地址的一个副本, 并不是原对象本身 。 所以对引用对象进行操作会同时改变原对象.  
一般认为,java内的传递都是值传递.

1. 请解释Java中的概念，什么是构造函数？什么是构造函数重载？

当新对象被创建的时候，构造函数会被调用。每一个类都有构造函数。在程序员没有给类提供构造函数的情况下，Java编译器会为这个类创建一个默认的构造函数。  
Java中构造函数重载和方法重载很相似。可以为一个类创建多个构造函数。每一个构造函数必须有它自己唯一的参数列表。

1. 请你讲讲abstract class 和 interface有什么区别

声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义static final成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口。

1. 请说明ArrayList和LinkedList的区别？

ArrayList和LinkedList都实现了List接口，他们有以下的不同点：  
ArrayList是基于索引的数据接口，它的底层是数组。它可以以O(1)时间复杂度对元素进行随机访问。与此对应，LinkedList是以元素列表的形式存储它的数据，每一个元素都和它的前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是O(n)。  
相对于ArrayList，LinkedList的插入，添加，删除操作速度更快，因为当元素被添加到集合任意位置的时候，不需要像数组那样重新计算大小或者是更新索引。  
LinkedList比ArrayList更占内存，因为LinkedList为每一个节点存储了两个引用，一个指向前一个元素，一个指向下一个元素。

1. 请说明List，Map，Set三个接口存取元素时，各有什么特点？

List以特定索引来存取元素，可以有重复元素。Set不能存放重复元素（用对象的equals()方法来区分元素是否重复）。Map保存键值对（key-value pair）映射，映射关系可以是一对一或多对一。Set和Map容器都有基于哈希存储和排序树的两种实现版本，基于哈希存储的版本理论存取时间复杂度为O(1)，而基于排序树版本的实现在插入或删除元素时会按照元素或元素的键（key）构成排序树从而达到排序和去重的效果。

1. 请说明一下垃圾回收的优点以及原理

Java 语言中一个显著的特点就是引入了垃圾回收机制，使c++程序员最头疼的内存管理的问题迎刃而解，它使得Java程序员在编写程序的时候不再需要考虑内存管理。由于有个垃圾回收机制，Java中的对象不再有"作用域"的概念，只有对象的引用才有"作用域"。垃圾回收可以有效的防止内存泄露，有效的使用可以使用的内存。垃圾回收器通常是作为一个单独的低级别的线程运行，不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收，程序员不能实时的调用垃圾回收器对某个对象或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收，增量垃圾回收。

**C/C++语法和基础知识**

1.说一下static关键字的作用

2.sizeof关键字的作用

int a;  sizeof(a)为4；

int \*p;  sizeof(p)为4；

int b[10]; sizeof(b)为数组的大小，4\*10；

3.const char crr[] = “123”和 char drr[] = “123”有什么区别

const char crr[] = "123";

//这里123本来是在栈上的，但是编译器可能会做某些优化，将其放到常量区

char drr[] = "123";

//字符串123保存在栈区，可以通过drr去修改

4.如何实现动态数组

5.写一个标准宏，这个宏输入两个参数并返回较小的一个

*#define MIN(x, y) ((x)<(y)?(x):(y))*

*6.数组指针和指针数组的区别*

*7.如何实现队列*

*8.修饰符extern的作用*

9.请你说一下函数指针

1、定义

函数指针是指向函数的指针变量。

函数指针本身首先是一个指针变量，该指针变量指向一个具体的函数。这正如用指针变量可指向整型变量、字符型、数组一样，这里是指向函数。

C在编译时，每一个函数都有一个入口地址，该入口地址就是函数指针所指向的地址。有了指向函数的指针变量后，可用该指针变量调用函数，就如同用指针变量可引用其他类型变量一样，在这些概念上是大体一致的。

1. 用途：

调用函数和做函数的参数，比如回调函数

10.值传递，地址传递和引用传递

**数据结构和算法**

1. 数组和链表的区别
2. 栈结构和队列的区别
3. 判断一个链表是否有环
4. 简述哈希表，什么是哈希冲突，项目中用过哪些哈希表
5. 二叉树的三种遍历的区别
6. 简述快速排序的过程
7. 冒泡排序优化
8. 树的数据结构可以通过什么样的方式在代表中实现
9. 什么是完全二叉树
10. 什么是平衡二叉树
11. 什么是堆排序，优先队列是什么，怎么实现优先队列
12. 如何使链表逆序
13. 如何获取单链表的长度
14. 如何检查两个字符串是否互相颠倒
15. 贪心算法和动态规划的区别
16. 图可以用什么养的形式来表现

**计算机网络和操作系统**

1. TCP和UDP之间的异同
2. TCP和UDP分别在实际代码中的应用，为什么会选择TCP/UDP；
3. HTTP协议和TCP之间存在什么样的关系
4. 什么是P2P
5. 三次握手和四次挥手分别是什么
6. 什么是帧同步
7. 什么是状态同步
8. 线程和进程的区别
9. 进程间通信的几种方式
10. 什么是缓冲区溢出？有什么危害？其原因是什么

**设计模式和软件架构**

1. 介绍一种你所了解的设计模式
2. 使用设计模式的好处有哪些
3. 状态模式和策略模式在UML上看是一样的，那么他们的区别是什么
4. 什么是MVC模式，MVC分别代表什么意思，为什么要有MVC；
5. 依赖注入是什么意思；
6. IOC容器是做什么的，实现依赖注入必须有IOC容器吗；
7. 单元测试的测试覆盖率是什么意思
8. 编写单元测试的好处有哪些
9. 什么是TDD（测试驱动开发）

**项目和游戏开发经验**

1. 简述下GameObject和Component
2. 简述下材质球
3. 什么是Prefab 使用Prefab的好处
4. OnColliderEnter和OnTriggerEnter区别
5. 简述下Transform的Translate方法和CharacterController的Move方法
6. Camera组件投影方式透视和正交的区别
7. Unity图片在不同平台下的压缩格式一般使用哪些
8. SVN和GIT的区别
9. GIT暂存区的作用
10. 什么是“树冲突”，产生的原因是什么，如何解决
11. 持续集成是什么