

3.Java 有几种基本数据类型？分别占几个字节？

数据类型	字节	默认值
byte	1	0
short	2	0
int	4	0
long	8	0
float	4	0.0f
double	8	0.0d
char	2	'\u0000'
boolean	4	false

4.有两个有序数组 **a[10, 20, 30]**, **b[5, 15, 25]**, 设计算法如何取出两个数组中第 3 小的数（也就是取出 15）；

设计时一定要考虑效率和内存问题。

5.HashMap 底层的原理，put(key, value)、get(key)原理

1、HashMap 的底层源码实现 put 方法：

现根据 key 的 hashCode（计算 hash 值的方法：`int hash = hash(key.hashCode())`），此方法加了高位运算，以防止 hash 冲突）重新计算 hash 值，然后再根据该 hash 值得到这个元素在数组中的位置（得到该 hash 值所对应 table 中索引的方法：`int i = indexFor(hash, table.length)`）（即下标）。如果该位置上没有元素，就直接将该元素放到此数组中的该位置上。若该位置上已经存放了其他元素了，则在该位置上的元素将以链表的形式存放，新加入的放在链表头，最先加入的放在链表尾。（把元素放到该索引位置上的方法：`addEntry(hash, key, value, i)`）。

2、HashMap 的底层源码实现 get 方法：

从 HashMap 中 get 元素时，首先根据 key 的 hashCode() 计算 key 的 hash 值（`int hash = hash(key.hashCode())`），找到数组中对应位置的某一元素（`table[i] =`

`indexOf(hash,table.length))`，然后通过 `key` 的 `equals` 方法在对应位置的链表中找到需要的元素的值。

6.接口与抽象类的区别

从设计层面来说，抽象是对类的抽象，是一种模板设计，接口是行为的抽象，是一种行为的规范。

Java 提供和支持创建抽象类和接口。它们的实现有共同点，不同点在于：接口中所有的方法隐含的都是抽象的。而抽象类则可以同时包含抽象和非抽象的方法。

类可以实现很多个接口，但是只能继承一个抽象类

类可以不实现抽象类和接口声明的所有方法，当然，在这种情况下，类也必须得声明成是抽象的。

抽象类可以在不提供接口方法实现的情况下实现接口。

Java 接口中声明的变量默认都是 `final` 的。抽象类可以包含非 `final` 的变量。

Java 接口中的成员函数默认是 `public` 的。抽象类的成员函数可以是 `private`，`protected` 或者是 `public`。

接口是绝对抽象的，不可以被实例化。抽象类也不可以被实例化，但是，如果它包含 `main` 方法的话是可以被调用的。

也可以参考 JDK8 中抽象类和接口的区别。

7.什么是抽象、封装、继承与多态

抽象：提取现实世界中某事物的关键特性，为该事物构建模型的过程。对同一事物在不同的需求下，需要提取的特性可能不一样。得到的抽象模型中一般包含：属性（数据）和操作（行为）。这个抽象模型我们称之为类。对类进行实例化得到对象。

封装：封装可以使类具有独立性和隔离性；保证类的高内聚。只暴露给类外部或者子类必须的属性和操作。类封装的实现依赖类的修饰符（`public`、`protected` 和 `private` 等）

继承：对现有类的一种复用机制。一个类如果继承现有的类，则这个类将拥有被继承类的所有非私有特性（属性和操作）。这里指的继承包含：类的继承和接口的实现。

多态：多态是在继承的基础上实现的。多态的三个要素：继承、重写和父类引用指向子类对象。父类引用指向不同的子类对象时，调用相同的方法，呈现出不同的行为；就是类多态特性。多态可以分成编译时多态和运行时多态。

8.常用的设计模式并分别讲解一下

<https://blog.csdn.net/wmq880204/article/details/75106848>

9.Linux 常用命令

<https://www.cnblogs.com/q1ng/p/4474527.html>

10.数据库的联合索引

11.数据库事务，ACID 的介绍

数据库事务正确执行的四个原则：A 原子性、C 一致性、I 独立性、D 持久性

（1）原子性：整个事务中的所有操作，要么全部完成，要么全部不完成，不可能停滞在中间某个环节。事务在执行过程中发生错误，会被回滚到事务开始前的状态，就像这个事务从来没发生过一样。

例如：A 账户中有 1000 元，B 账户中有 1000 元。A 要给 B 转账 500 元。A 扣款和 B 加款这两条要么同时执行，要么同时不执行。如果在 A 扣款后 B 加款之前，系统发生故障，会回滚到 A 扣款之前的状态。

（2）一致性：事务开始之前和事务结束后，数据库的完整性约束没有被破坏。

例如：不论汇款成败，A 账户 B 账户总额是 2000 元。

（3）独立性：事务的执行互不干扰。

影响有以下几种可能：

脏读：一个事务误读了另一个事务未提交的数据，而这个数据可能回滚。

不可重复读：一个事务范围内的两个相同查询返回了不同数据。如：事务 A 对某个查询执行了两次，当第一次执行完时，事务 B 对某数据进行了修改，事务 A 再次查询时，数据发生了变化。

幻读（虚读）：例如事务 A 对一个表中的全部行进行了修改，事务 B 向表中插入了一行，事务 A 发现表中还有没有修改的数据，仿佛出现了幻觉。

丢失更新：两个事务同时读取一条记录，事务 A 先修改记录，事务 B 也修改记录（B 不知道 A 修改过），B 提交数据后，B 的结果覆盖了 A 的修改结果。

串行执行事务可以避免这些问题，但并发执行能显著改善性能。可通过锁阻塞的方式解决上

述问题。

(4) 持久性：事务执行成功后，该事务对数据库的更改是持久保存在数据库中的，不会被回滚。

可以使用日志记录或影子副本来实现。

12.从坐标原点（0， 0）到坐标（m， n）有多少条路径

```
int fun(int m,int n)
{
    if(m==0&& n==0)
    {
        return 0;
    }
    else if(m==0||n==0)
        return 1;
    else
        return (fun(m-1,n)+fun(m,n-1));
}
```

13.Java 虚拟机的堆与栈以及回收算法有哪些？新生代用什么算法，老生代用什么算法？

<https://www.cnblogs.com/huajiezh/p/5769255.html>

14.你有什么想问我的吗？可以随便问。

15.平时关注谁的博客？有 GitHub 项目吗？