**7-9模拟面试题目和答案**

**第二组：**

**1、**抽象类能否使用 final 声明？ · 不能，因为 final 属于终结器，被其修饰的类是不能有子类的 · 而抽象类本身必须有子类，所以不能。

2、多态的前提：

1.要有继承关系。

2.要有方法重写。

3.要有父类引用指向子类对象。

**第三组：**

1. 数组扩容：

public static int[] addArray(int[] arr1) {

int[] arr2=new int[arr1.length\*2];//新数组长度

for (int i = 0 ; i < arr1.length ; i++) {

arr2[i] = arr1[i];

}

return arr2;

}

2、private 声明的变量及方法，只在声明的类内可以使用。public声明的变量及方法，表明在整个包内包外都可使用。

**第四组：**

1. 移位运算，

左移：整体左移，低位全部补0。

带符号右移：整体右移，高位全部补符号数。

无符号右移：最高位全部补0.

2、方法的重载，方法名相同，参数列表不同，即参数类型或者参数个数不同。

**第五组：**

1. 构造方法：格式：名称与类名相同，无返回类型的声明。
2. 面向对象特征：封装性、继承性、多态性、抽象性

**第六组**

public class JavaIntegerCache {

public static void main(String... strings) {

Integer integer1 = 3;

Integer integer2 = 3;

if (integer1 == integer2)

System.out.println("integer1 == integer2");

else

System.out.println("integer1 != integer2");

Integer integer3 = 300;

Integer integer4 = 300;

if (integer3 == integer4)

System.out.println("integer3 == integer4");

else

System.out.println("integer3 != integer4");

}

}

2、在Java 5中，在Integer的操作上引入了一个新功能来节省内存和提高性能。整型对象通过使用相同的对象引用实现了缓存和重用。适用于整数值区间-128 至 +127。只适用于自动装箱。使用构造函数创建对象不适用。 缓存通过一个for循环实现。从低到高并创建尽可能多的整数并存储在一个整数数组中。这个缓存会在Integer类第一次被使用的时候被初始化出来。以后，就可以使用缓存中包含的实例对象，而不是创建一个新的实例(在自动装箱的情况下)。 这个功能在Java 5中引入的时候,范围是固定的-128 至 +127。后来在Java 6中，可以通过java.lang.Integer.IntegerCache.high设置最大值。

**第七组**

1. Java中所有类的祖宗类是Object
2. final的关键字 用final修饰变量、修饰方法、修饰类会有什么结果？ 用final修饰变量，表示常量，内容不可被修改 用final修饰方法，该方法不能被子类重写 用final修饰类，该类不能被继承
3. 抽象类和接口的区别？

1、抽象类使用abstract修饰；

1. 抽象类不能实例化，即不能使用new关键字来实例化对象；
2. 含有抽象方法（使用abstract关键字修饰的方法）的类是抽象类，必须使用abstract关键字修饰；
3. 抽象类可以含有抽象方法，也可以不包含抽象方法，抽象类中可以有具体的方法；
4. 如果一个子类实现了父类（抽象类）的所有抽象方法，那么该子类可以不必是抽象类，否则就是抽象类；
5. 抽象类中的抽象方法只有方法体，没有具体实现
6. 接口使用interface修饰；
7. 接口不能被实例化；
8. 一个类只能继承一个类，但是可以实现多个接口；
9. 接口中方法均为抽象方法；
10. 接口中不能包含实例域或静态方法（静态方法必须实现，接口中方法是抽象方法，不能实现）
11. 抽象类和普通类的两个最大区别：
12. 抽象类不能被实例化，就是不能用new调出构造方法创建对象,而普通类则反之。

2.抽象的修饰的方法不可以有方法体，天生是被重写的，抽象方法只可以定义在抽象类中