

JAVASE面试题集

根底篇

JavaSE根底局部

Q1: 简述equals() 和 ==之间的区别?

A: Object类中的equals()比较的是两个对象的地址。

String、Integer类中的equals()比较的是两个对象所指的内容。

== 是运算符，比较的是两个变量的类型和值是否相等。

JavaSE根底局部

Q2: 简述方法重载`overload`和方法覆盖`override`的区别?

A: 方法的重写`Overrid`和重载`Overload`是Java多态性的不同表现。重写是父类与子类之间多态性的一种表现，重载是一个类中多态性的一种表现。

如果在一个类中，存在着这样一组方法：方法名相同，参数表不同（类型不同，个数不同，顺序不同），与返回值类型无关，这就是方法重载。

如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写。

JavaSE根底局部

Q3: 简述方法重写的规那么？

A: 1.子父类方法名相同，参数表相同

2.子父类方法的返回值类型要相同

【如果返回值存在子父类，子类方法的返回值范围不能大于父类】

3.子类方法的修饰符范围必须大于等于父类方法的修饰符范围

【private、final、static修饰的方法不构成覆盖】

4.子类方法声明的异常范围必须小于等于父类方法声明异常的范围

JavaSE根底局部

Q4: 抽象类是否有构造方法？接口是否有构造方法？

A: 抽象类可以有构造方法，但不可以有抽象的构造方法。

接口是纯抽象类，没有构造方法

Q: 既然抽象类可以有构造方法为什么又不能被实例化？

A: 抽象类的构造方法不是为了实例化自身，而是为了方便子类调用。

JavaSE根底局部

Q5: Java中的类为什么只能单继承？接口为什么可以多实现？

A: 假设为多继承，那么当多个父类中有重复的属性或者方法时，子类的调用结果会模糊不清，因此只能单继承。

通过实现接口拓展了类的功能，假设实现的多个接口中有重复的方法也没关系，因为实现类中必须重写接口中的方法，所以调用时还是调用的实现类中重写的方法。

JavaSE根底局部

Q6: 简述final、finally、和finalize的用法?

A: **final**修饰成员变量必须赋初值，并且不可改变值，相当于定义一个常量，**final**修饰成员方法，该方法不能被覆盖，**final**修饰类该类不能被继承，**final**是唯一一个能修饰局部变量的修饰符。

finally是异常机制中用来提供绝对能够被执行处理的代码块。如：流的关闭操作、数据库连接的关闭操作等。

finalize方法是Object类中的方法，是JVM垃圾回收时调用的方法，该方法人为调用无效。

JavaSE根底局部

Q7: 说出5个常见的运行时异常 (RuntimeException) ?

A: ArithmeticException 数学异常 (除数为0)
NullPointerException 空指针引用异常
ClassCastException 类的造型异常
IllegalArgumentException 非法参数传递异常
IndexOutOfBoundsException 索引越界异常

JavaSE根底局部

Q8: 简述GC垃圾回收机制？程序员如何人为释放内存？

A: Java程序中，**new** 出来的东西都是驻留在堆(heap)内，而栈(stack)里空间调用完都会自动释放，而堆里不是，像C和C++都要人为的去主动释放，而JAVA那么由JVM来处理垃圾回收，JVM何时释放堆内存，只跟内存有关，当内存缺乏时，JVM会调用**finalize()**方法释放堆内存，人为调用**finalize()**方法无效。

人为释放内存可以调用以下两个方法：

1.`Runtime.getRuntime().gc();`

2.`System.gc();`

JavaSE根底局部

Q9: 简述集合List、Set、Map的区别？

A: List和Set是Collection的子类，而Map不是。

List关心的是索引，可以放重复的对象，输出的顺序和放入的顺序一致。

Set关心的是唯一性，不可以放重复的对象，有自己的一套排序规那么。

List与Set只能存储单一的对象，而Map可以存储成对的对象，使用key-value来映射和存储数据，Key必须惟一，value可以重复。

JavaSE根底局部

Q10: 集合Map是否有迭代器？如何迭代？

A: Map集合中没有迭代器，需要把Map转换成Set来进行迭代。Map中提供了两个方法：

1.keySet()方法是把Map中的key转换成一个Set集合，通过Set集合中的迭代器Iterator迭代key时，根据key获取value的值。

2.entrySet() 方法是把Map中的key和value看成是一个整体转换成一个Set<Map.Entry<K,V>>集合，通过Set集合中的迭代器迭代出key和value值。

JavaSE根底局部

Q11: TreeSet集合中是否可以放入多个同类型的对象？

A: TreeSet集合是按自然排序规那么来排序，当放入多个同类对象时，需要看这些对象是否满足排序规那么，如果不满足排序规那么，那么会报ClassCastException异常。可以用Comparable接口或Comparator接口为集合提供自己的排序规那么。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/916105221051010111>