

Java基础

1. JDK和JRE的区别？

JDK全称Java Development Kit，是功能齐全的Java SDK，包含编译器javac和各类工具（javadoc、jdb等等），能够创建和编译程序。

JRE全称Java Runtime Environment，是运行已编译Java程序所需组建的集合，包括JVM、Java类库、Java命令等，不能够穿件和编译程序。

2. == 和 equals的区别？

==: 判断对象的地址（本质也是比较值，比较地址值是否相同）是否相同，对基本数据类型是判断值

equals: 只能用于判断对象是否相等，若重写过则是比较对象内容（如String）

3. 两个对象的 hashCode()相同，则 equals()也一定为 true，对吗？

错，首先hashCode在散列表如HashSet中才有用，主要是通过散列码来计算出对象在散列表中的位置，因为计算hashCode的算法会导致碰撞，也就是哈希冲突，导致一个hashCode会对应多个对象，所以在HashSet中对相同HashCode的对象还要调用equals判断对象是否真的相等。

对象想等则hashCode一定相等，反之不一定。故该说法错误。

4. final static this super 关键字在 java 中有什么作用？

final: 主要用来标志无法修改。

- 修饰的类无法被继承，且该类中所有方法隐式声明为final方法；
- 修饰的方法无法被override； -
- 修饰的变量是常量，基本数据类型初始化后无法再更改，引用类型无法指向其他对象。

static:

- 修饰成员变量则该变量在驻留内存中，被类中所有对象共享，修饰成员方法则该方法是静态方法，存放在方法区，调用的时候是Class.variable和Class.Method；
- 修饰静态代码块，则改代码块在该类初始化时执行且只执行一次，在非静态代码块之前执行(静态代码块—>非静态代码块—>构造方法)；
- 静态内部类：不依赖外围类的创建就可以创建（new OuterClass.InnerClass），编译完成后无法访问外围类的非static变量和方法；
- 静态导包：import static xxxx.xxxx;

this: 用来指向引用类的当前实例（有时候可以省略）

super: 用于从子类访问父类的变量和方法

5. java 中操作字符串都有哪些类？它们之间有什么区别？

- String: final类型，无法更改，每次操作String类型数据都会生成新的对象
- StringBuilder: 线程不安全，用的最多，主要有append, delete, insert等常规操作
- StringBuffer: 操作都加了synchronized锁，线程安全，性能比SB略差

6. String str="i"与 String str=new String("i")一样吗？

不一样。先说第一种的过程，现在字符串常量池检查有没有"i"，有则把str引用指向它，没有则创建一个再指向；第二种是直接堆中创建了一个新的值为i的String对象。（池中有了则只在堆中创建一个对象，没有则先池再堆分别创建两个对象）

7. 抽象类必须要有抽象方法吗？普通类和抽象类有哪些区别？

不一定。主要有三点区别：

- 抽象类默认方法为public，可声明为protected、default，不能为私有因为抽象类就是用来继承的；
- 抽象类只能被继承不能创建对象；
- 抽象类中抽象方法必须被子类重写，否则继承抽象类的子类也必须声明为抽象类

8. 接口和抽象类的区别？

- 首先本质上接口不属于类(属于引用类型)，接口方法默认public且所有方法不能有实现(java8之后可以有实现)，抽象类则可以有实现；
- 接口中只允许有static和final变量，抽象类不一定；
- 一个类可以实现多个接口但只能继承一个抽象类（java里不允许多继承），接口本身还可以extend多个接口；
- 设计层面，抽象是针对类的抽象，接口是针对行为（方法）的抽象

9. java 中 IO 流分为几种？

- 流向：InputStream(byte)/Reader(char)和OutputStream/Writer
- 操作单元：字节流和字符流
- 流角色：节点流和处理流

10. BIO,NIO,AIO 有什么区别？

BIO：Blocking I/O，同步阻塞I/O模式，数据的读写必须阻塞在一个线程内等待其完成。优点在于编程模型简单，不用考虑负载、限流等问题，适用于 活动连接数不高的情况，缺点在于面对上万级的高并发量连接的IO请求时无能为力。

NIO：(Non-blocking/New I/O) NIO 是一种同步非阻塞的 I/O 模型，提供了Channel、Selector、Buffer等抽象，支持基于面向缓冲基于通道的I/O操作方法。提供了SocketChannel和ServerSocketChannel两种不同的套接字通道实现，支持阻塞和非阻塞模式。根据负载和并发的情况选择NIO模式来开发。

AIO：(Asynchronous I/O) AIO 是异步非阻塞的 IO 模型，基于事件和回调机制实现，也就是应用操作之后会直接返回不会阻塞。