

第1章 Java核心基础面试题汇总

摘要：本章内容，享学课堂在Java筑基中有系统化-全面完整的直播讲解，详情加微信：xxgfwx03

第1章 Java核心基础面试题汇总

1.1 抽象类和接口的区别~Leo

这道题想考察什么？

考察的知识点

考生应该如何回答

抽象类

接口

抽象类和接口的区别

1.2 重载和重写的区别（京东）~jett

这道题想考察什么？

考察的知识点

考生应该如何回答

1.3 什么是值传递和引用传递，Java 是值传递还是引用传递？~derry

这道题想考察什么？

考察的知识点

考生应该如何回答

1.什么是值传递和引用传递，Java 是值传递还是引用传递？

1.4 String s = new String("");创建了几个对象？~jett

这道题想考察什么？

考察的知识点

考生应该如何回答

1.5 java里 equals和== 区别~leo

这道题想考察什么？

考察的知识点

考生应该如何回答

java当中的数据类型和“==”的含义

equals()方法介绍

String类的equals()方法和==的区别

结论

1.6 try-catch-finally, try里有return, finally还执行么？~jett

这道题想考察什么？

考察的知识点

考生应该如何回答

1.7 Exception与Error区别~jett

这道题想考察什么？

考察的知识点

考生应该如何回答

1.8 Static class 与non static class的区别~jett

这道题想考察什么？

考察的知识点

考生应该如何回答

1.1 抽象类和接口的区别~Leo

这道题想考察什么？

java是面向对象编程的，抽象是它的一大特征，通过这道题可以考察同学们对封装抽象层的理解

考察的知识点

考察抽象类和接口封装的区别

考生应该如何回答

从下面三个方面回答。

抽象类

首先我们来看下什么是抽象方法：

```
abstract void fun();
```

抽象方法必须用abstract关键字进行修饰。如果一个类含有抽象方法，则称这个类为抽象类，抽象类必须在类前用abstract关键字修饰。因为抽象类中含有无具体实现的方法，所以不能用抽象类创建对象。

```
[public] abstract class ClassName {  
    abstract void fun();  
}
```

从这里可以看出，抽象类就是为了继承而存在的，如果你定义了一个抽象类，却不去继承它，那么等于白白创建了这个抽象类，因为你不能用它来做任何事情。对于一个父类，如果它的某个方法在父类中实现出来没有任何意义，必须根据子类的实际需求来进行不同的实现，那么就可以将这个方法的声明为abstract方法，此时这个类也就成为abstract类了。

包含抽象方法的类称为抽象类，但并不意味着抽象类中只能有抽象方法，它和普通类一样，同样可以拥有成员变量和普通的成员方法。注意，抽象类和普通类主要有以下三方面区别：

- 1) 抽象方法必须为public或者protected（因为如果为private，则不能被子类继承，子类便无法实现该方法），缺省情况下默认为public。
- 2) 抽象类不能用来创建对象；
- 3) 如果一个类继承于一个抽象类，则子类必须实现父类的抽象方法。如果子类没有实现父类的抽象方法，则必须将子类也定义为abstract类。

接口

接口是对行为的抽象，通过如下方式定义一个接口：

```
[public] interface InterfaceName {  
  
}
```

接口中可以含有变量和方法。但是要注意，接口中的变量会被隐式地指定为public static final变量（并且只能是public static final变量，用private修饰会报编译错误），而方法会被隐式地指定为public abstract方法且只能是public abstract方法（用其他关键字，比如private、protected、static、final等修饰会报编译错误），并且接口中的所有方法不能有具体的实现，也就是说，接口中的方法必须都是抽象方法。

抽象类和接口的区别

语法层面上的区别：

- 1) 抽象类可以提供成员方法的实现细节，而接口中只能存在public abstract 方法；
- 2) 抽象类中的成员变量可以是各种类型的，而接口中的成员变量只能是public static final类型的；
- 3) 接口中不能含有静态代码块以及静态方法，而抽象类可以有静态代码块和静态方法；
- 4) 一个类只能继承一个抽象类，而一个类却可以实现多个接口。

设计层面上的区别：

- 1) 抽象类是对一种事物的抽象，即对类抽象，而接口是对行为的抽象。抽象类是对整个类整体进行抽象，包括属性、行为，但是接口却是对类局部（行为）进行抽象。例如：超人和鸟是不同类的事物，但是它们都有一个共性，就是都会飞。那么在设计的时候，可以将超人和鸟涉及为类，但是不能将飞行设计为类，因此它只是一个行为特性，并不是对一类事物的抽象描述。超人和鸟都可以实现这个接口，所以，继承是一个“是不是”的关系，而接口实现则是“有没有”的关系。
- 2) 设计层面不同，抽象类作为很多子类的父类，它是一种模板式设计。而接口是一种行为规范，它是一种辐射式设计。

1.2 重载和重写的区别（京东）~jett

这道题想考察什么？

java基础

考察的知识点

面向对象基础知识

考生应该如何回答

重写与重载之间的区别

方法重载：在同一个类中，允许存在多个同名方法，只要它们的参数列表不同，与修饰符和返回值类型无关，这样的多个同名方法之间构成重载。

重载的特点是：

- 1、同一个类中
- 2、方法名相同，参数列表不同（参数顺序、个数、类型）
- 3、方法返回值、访问修饰符任意
- 4、与方法的参数名无关
- 5、如LayoutInflater中的inflate(...)方法

方法重写：当父类中的方法对于子类来说不适用时，子类可以对从父类中继承来的方法进行重写，用于增强父类中方法的功能

重写的特点是：

- 1、有继承关系的子类中

- 2、方法名相同，参数列表相同（参数顺序、个数、类型），方法返回值相同
- 3、访问修饰符，访问范围需要大于等于父类的访问范围
- 4、与方法的参数名无关
- 5、如Activity类的承继体系中的protected void onCreate(Bundle savedInstanceState)方法

1.3 什么是值传递和引用传递，Java 是值传递还是引用传递？~derry

这道题想考察什么？

1. 是否了解什么是值传递和引用传递与真实场景使用，是否熟悉什么是值传递和引用传递在工作中的表现是什么？

考察的知识点

1. 什么是值传递和引用传递的概念在项目中使用与基本知识

考生应该如何回答

1.什么是值传递和引用传递，Java 是值传递还是引用传递？

答：

当一个对象被当作参数传递到一个方法后，在此方法内可以改变这个对象的属性，那么这里到底是值传递还是引用传递？

答：是值传递。Java 语言的参数传递只有值传递。当一个实例对象作为参数被传递到方法中时，参数的值就是该对象的引用的一个副本。指向同一个对象，对象的内容可以在被调用的方法内改变，但对象的引用(不是引用的副本)是永远不会改变的。

Java的参数传递，不管是基本数据类型还是引用类型的参数，都是按值传递，没有按引用传递！

对于参数的传递可以分为两种情况：

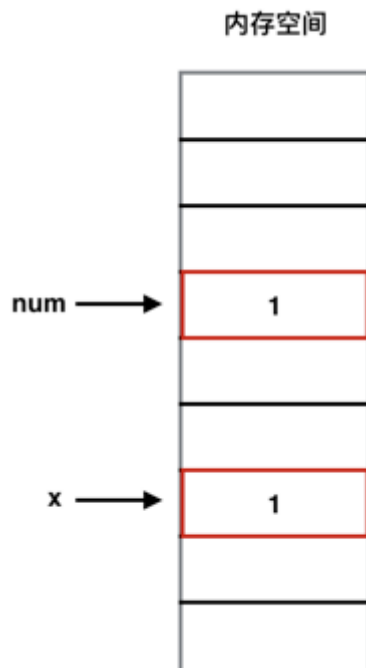
1.基本数据类型的参数

```
1 public class TransferTest {
2     public static void main(String[] args) {
3         int num = 1;
4         System.out.println("changeNum()方法调用之前: num = " + num);
5         changeNum(num);
6         System.out.println("changeNum()方法调用之后: num = " + num);
7     }
8
9     public static void changeNum(int x) {
10        x = 2;
11    }
12 }
```

运行结果：

```
changeNum()方法调用之前: num = 1
changeNum()方法调用之后: num = 1
```

传递过程的示意图如下:



分析: num作为参数传递给changeNum()方法时,是将内存空间中num所指向的那个存储单元中存放的值1复制了一份传递给了changeNum()方法中的x变量,而这个x变量也在内存空间中分配的一个存储单元。这时就把num对的值1传递给了x变量所指向的存储单元中。此后在changeNum()方法中对x变量的一切操作都是针对于x所指向的这个存储单元,与num所指向的存储单元无关。

所以,在changeNum()方法被调用后,num所指向的存储单元的值还是没有发生变化,这就是所谓的“值传递”。

值传递的精髓是:传递的是存储单元中的内容,而不是存储单元的引用。

2. 引用类型的参数

```
1 public class TransferTest2 {
2     public static void main(String[] args) {
3         Person person = new Person();
4         System.out.println(person);
5         change(person);
6         System.out.println(person);
7     }
8
9     public static void change(Person p) {
10        p = new Person();
11    }
12 }
13
14 /**
15  * Person类
16  */
```

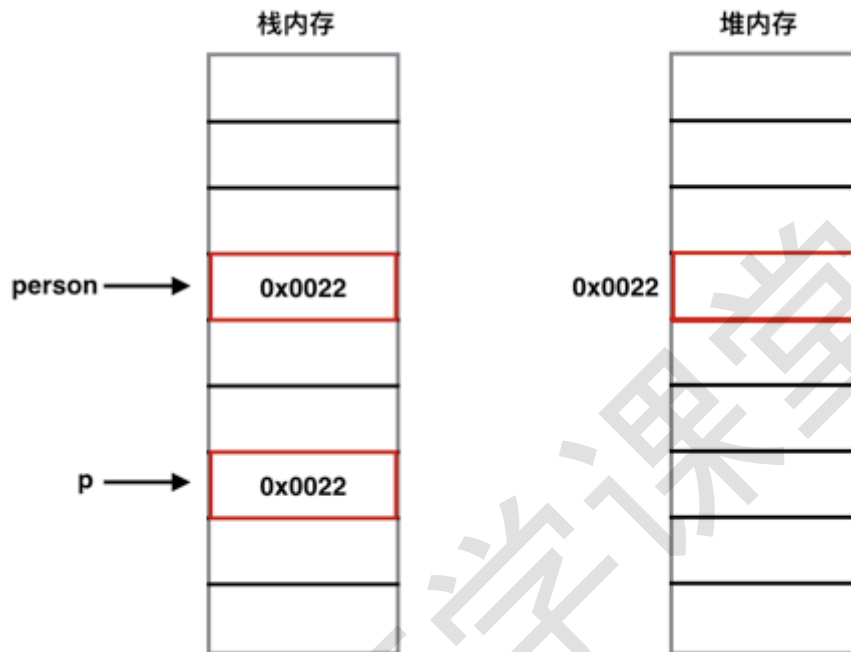
```
17 class Person {
18
19 }
```

运行结果：

```
Person@1d44bcfa
Person@1d44bcfa
```

可以看出两次打印结果一致。即调用change()方法后，person变量并没发生改变。

传递过程的示意图如下：



分析：

01.当程序执行到第3行 `Person person = new Person()`时，程序在堆内存（heap）中开辟了一块内存空间用来存储Person类实例对象，同时在栈内存（stack）中开辟了一个存储单元来存储该实例对象的引用，即上图中person指向的存储单元。

02.当程序执行到第5行 `change(person)`时，person作为参数（实参）传递给change()方法。这里是person将自己的存储单元的内容传递给了change()方法的p变量。此后在change()方法中对p变量的一切操作都是针对于p变量所指向的存储单元，与person所指向的存储单元就没有关系了。

1.4 String s = new String("");创建了几个对象?~jett

这道题想考察什么？

java基础

考察的知识点

是否看过String源码

考生应该如何回答

这行代码转成class字节码

```
new          #2          // class java/lang/String
dup
ldc          #3          // String
invokespecial #4          // Method java/lang/String.<init>:
(Ljava/lang/String;)V
astore       4
```

我们可以看到new了一个string对象之后会调用构造方法
同时从String类源码中可以看到

```
public String(String original) {
    // Android-changed: Constructor unsupported as all calls are intercepted by
    the runtime.
    throw new UnsupportedOperationException("Use StringFactory instead.");
}
```

这里也new了一个UnsupportedOperationException对象

同时String中还定义了一个数组如下：

```
private static final ObjectStreamField[] serialPersistentFields =
    new ObjectStreamField[0];
```

所以答案是创建了3个对象

1.5 java里 equals和== 区别~leo

这道题想考察什么？

考察对 equals 的实现以及栈和堆的内存管理。

考察的知识点

equals 的实现以及栈和堆的内存管理。

考生应该如何回答

java当中的数据类型和“==”的含义

- 基本数据类型（也称原始数据类型）：byte,short,char,int,long,float,double,boolean。他们之间的比较，应用双等号（==），比较的是他们的值。
- 引用数据类型：当他们用（==）进行比较的时候，比较的是他们在内存中的存放地址（确切的说，是堆内存地址）。

注：对于第二种类型，除非是同一个new出来的对象，他们的比较后的结果为true，否则比较后结果为false。因为每new一次，都会重新开辟堆内存空间。

equals()方法介绍

equals 方法是 Object 中定义的一个方法，源码如下：

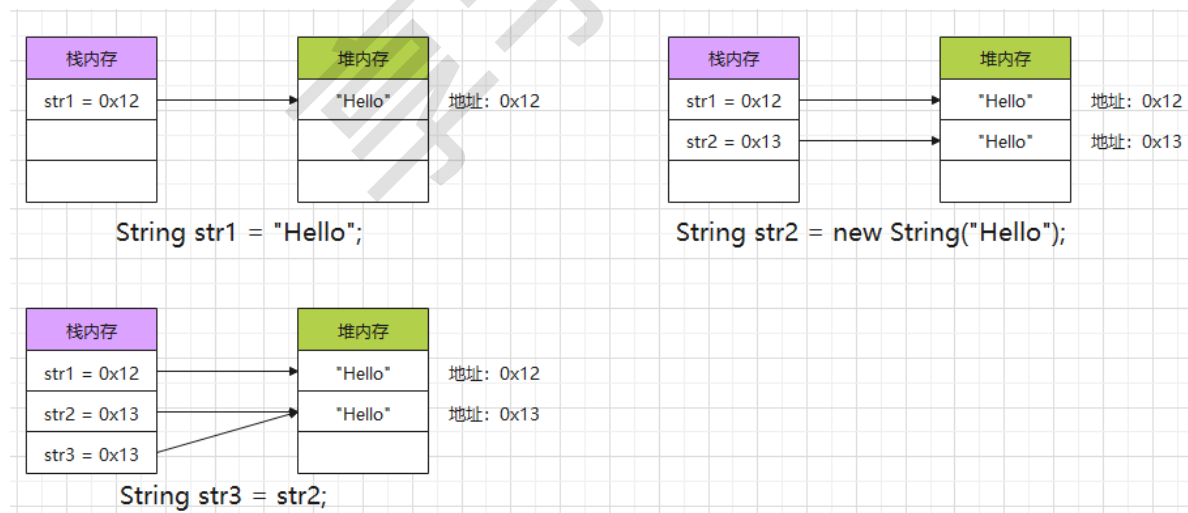
```
public boolean equals(Object obj) {  
    return (this == obj);  
}
```

可以看到 equals 实际上就是用的 == 实现的，所以这个原始方法意义不是很大，一般在类中需要做比较的时候，都会重写这个方法，如String、Integer、Date。

String类的equals()方法和==的区别

```
public class StringDemo {  
    public static void main(String args[]) {  
        String str1 = "Hello";  
        String str2 = new String("Hello");  
        String str3 = str2; // 引用传递  
        System.out.println(str1 == str2); // false  
        System.out.println(str1 == str3); // false  
        System.out.println(str2 == str3); // true  
        System.out.println(str1.equals(str2)); // true  
        System.out.println(str1.equals(str3)); // true  
        System.out.println(str2.equals(str3)); // true  
    }  
}
```

栈和堆的内存分析图：



由此可见，equals 是比较字符串的内容是否一样，== 是比较两个字符串的堆内存的地址是否一样。

结论

equals和==的区别，需要分两种情况讨论：

1. 没有重写 equals 方法，则 equals 和 == 是一样的。
2. 如果重写了 equals，则需具体看 equals 的方法实现。以 String 类为例：

1. equals 是比较字符串的内容是否一样。
2. == 是比较两个字符串的堆内存的地址是否一样，或者说引用的值是否相等。

1.6 try-catch-finally, try里有return, finally还执行么? ~jett

这道题想考察什么?

java基础

考察的知识点

java基本语法

考生应该如何回答

finally中的代码是会执行的。

[Java官方文档](#)上是这么描述的:

The `finally` block **always** executes when the `try` block exits.

该语法的设计就是为了在try,catch中不管出现任何情况都需要执行的代码，才会写入finally中。

[JVM规范](#)里面同样也有明确说明

If the try clause executes a return, the compiled code does the following:

1. Saves the return value (if any) in a local variable.
2. Executes a jsr to the code for the finally clause.
3. Upon return from the finally clause, returns the value saved in the local variable.

意思就是如果在try中return的情况下，先把try中将要return的值先存到栈帧的局部变量表中，接下来去执行finally语句块，最后再从局部变量表中取回return的值。

同时规范规定了，当try和finally里都有return时，会忽略try的return，而使用finally的return。

参考如下代码：

```
try{
    return;
}catch (Exception e){

}finally {
    int x=1;
}
```

查看对应的class字节码

```
//.....  
L3  
    LINENUMBER 14 L3  
    ICONST_1  
    ISTORE 2  
L4  
    LINENUMBER 10 L4  
    RETURN  
//.....
```

我们会看到finally中的 int x=1;先执行，而后是try中的return;

1.7 Exception与Error区别~jett

这道题想考察什么？

考察对java中异常和错误的理解

考察的知识点

java基础

考生应该如何回答

首先Exception和Error都是继承于Throwable 类，在Java 中只有 Throwable 类型的实例才可以被抛出（throw）或者捕获（catch），它是异常处理机制的基本组成类型。

Exception和Error体现了JAVA这门语言对于异常处理的两种方式。

Exception是java程序运行中可预料的异常情况，咱们可以获取到这种异常，并且对这种异常进行业务外的处理。

Error是java程序运行中不可预料的异常情况，这种异常发生以后，会直接导致VM不可处理或者不可恢复的情况。所以这种异常不可能抓取到，比如OutOfMemoryError、NoClassDefFoundError等。

其中的Exception又分为检查性异常和非检查性异常。两个根本的区别在于，**检查性异常** 必须在编写代码时，使用try catch捕获（比如：IOException异常）。**非检查性异常** 在代码编写使，可以忽略捕获操作（比如：ArrayIndexOutOfBoundsException），这种异常是在代码编写或者使用过程中通过规范可以避免发生的。切记，Error是Throw不是Exception。

1.8 Static class 与non static class的区别~jett

这道题想考察什么？

静态内部类和非静态内部类应用

考察的知识点

java基础

考生应该如何回答

根据Oracle官方的说法：

Nested classes are divided into two categories: static and non-static. Nested classes that are declared static are called static nested classes. Non-static nested classes are called inner classes.

从字面上看，一个被称为静态嵌套类，一个被称为内部类。

而我们一般称之为静态内部类和非静态内部类

static class 静态内部类

- 1、用static修饰的是内部类，这个内部类变为静态内部类；
- 2、内部静态类不需要有指向外部类的引用；
- 3、静态类只能访问外部类的静态成员，不能访问外部类的非静态成员；
- 4、开发中多数用于设计外部类与内部类关联非常紧密，同时又能独立使用的情况，如RxJava中的操作符map,concatMap等就和各种Observer绑定写在同一个文件中。

non static class 非静态内部类

- 1、非静态内部类需要持有对外部类的引用；
- 2、非静态内部类能够访问外部类的静态和非静态成员；
- 3、一个非静态内部类不能脱离外部类实体被创建；
- 4、一个非静态内部类可以访问外部类的数据和方法；
- 5、开发中多数用于设计外部类与内部类关联非常紧密，多数情况是使用匿名内部类在使用；
- 6、特别需要注意非静态内部类内存泄漏的问题。