1. 面向对象和面向过程的区别

面向过程是一种基础的方法，它考虑的是实际的实现

面向对象的方法主要是把事物给对象化，包括其属性和行为

面向过程是具体化的，流程化的。解决一个问题，需要一步一步分析需要怎样，然后需要怎样，一步一步实现的。面向对象是模型化的，抽象出一个类，一个封闭的环境，在这个环境中有数据有解决问题的方法，需要什么功能直接使用就可以了

1. 面向对象的特征有哪些方面

封装 继承 多态

封装：把过程和数据包围起来，对数据的访问只能通过已定义的方法 java实质上将属性私有化,在其他类中直接访问不到，设置公开的方法setXX/getXX方法对于属性赋值以及获取保证 数据安全性 保密性

继承：子类继承父类的特征和行为,实现代码重用

多态：同一种事物，由于条件不同，产生的结果也不同 ，同一个父类引用类型，使用不同的子类实例而执行不同操作

1. 什么是多态机制？Java 语言是如何实现多态的

多态：同一种事物，由于条件不同，产生的结果也不同 ，同一个父类引用类型，使用不同的子类实例而执行不同操作

Java中多态的实现方式：接口实现，继承父类进行方法重写，同一个类中进行方法重载。

多态存在的三个必要条件

一、要有继承；

二、要有重写；

三、父类引用指向子类对象。

1. 抽象类和接口的对比

抽象类：方法可以有具体的方法，属性没有限制，构造方法可以有可以没有，不能实例化，是单一继承的

接口：方法必须为抽象方法，jdk1.8后增加默认方法，属性是公开静态常量，多重实现，不能实例化，没有构造方法。

1. 普通类和抽象类有哪些区别？

抽象类不能被实例化，普通类可以实例化

抽象类可以有抽象方法，抽象方法只需申明，无需实现，普通类不可有抽象方法

1. 抽象类能使用 final 修饰吗？

不能,定义抽象类就是让其他类继承的,如果定义为 final 该类就不能被继承,这样彼此就会产生矛盾,所以 final 不能修饰抽象类,编辑器也会提示错误信息

1. 创建一个对象用什么关键字？对象实例与对象引用有何不同？

new关键字，new创建对象实例（对象实例在堆内存中），对象引用指向对象实例（对象引用存放在栈内存中）。一个对象引用可以指向0个或1个对象;一个对象可以有n个引用指向它；

1. 成员变量与局部变量的区别有哪些

1、定义位置的区别

成员变量定义在函数（方法）外面，类的里面。

局部变量定义在函数内部。

2、作用上的区别

成员变量表示类的公共属性。它的值的不同决定了每个对象的差异。针对整个类有效。

局部变量就是函数内的一个普通变量。是辅助这个函数完成任务的临时变量。只在某个范围内有效。(一般指的就是方法,语句体内)  
 3、生命周期的区别

成员变量随着对象的创建而存在。

局部变量是当调用该成员变量所在的方法时候产生的，方法执行结束，局部变量释放。

4、初始化的区别

成员变量的初始值：

int 0

float 0.0

double 0.0

char null

boolean false

String null

局部变量必须先声明才可以在该方法内使用。

5.使用原则

在使用变量时需要遵循的原则为：就近原则  
首先在局部范围找，有就使用；接着在成员位置找。

1. 在 Java 中定义一个不做事且没有参数的构造方法的作用

Java 程序在执行子类的构造方法之前，如果没有用 super() 来调用父类特定 的构造方法，则会调用父类中“没有参数的构造方法”。因此，如果父类中只定 义了有参数的构造方法，而在子类的构造方法中又没有用 super() 来调用父类 中特定的构造方法，则编译时将发生错误，因为 Java 程序在父类中找不到没 有参数的构造方法可供执行。解决办法是在父类里加上一个不做事且没有参数 的构造方法。

1. 在调用子类构造方法之前会先调用父类没有参数的构造方法，其目的是？

帮助子类做初始化工作

1. 一个类的构造方法的作用是什么？若一个类没有声明构造方法，程序能正确执行吗？为什么？

主要作用是完成对类对象的初始化工作。可以执行。因为一个类即使没有声明构造方法也会有默认的不带参数的构造方法。

1. 构造方法有哪些特性？

名字与类名相同；  
没有返回值，但不能用void声明构造函数；  
生成类的对象时自动执行，无需调用。

1. 静态变量和实例变量区别

静态变量前面要加static，实例变量不用

实例变量必须创建对象后才可以通过这个对象来使用，静态变量则可以直接使用类名来引用。

1. 静态变量与普通变量区别

静态变量是在堆分配的，而普通变量是在栈上分配的，栈上分配的变量是动态的，可以实现重用，而堆上分配的变量是不能实现重用。还有普通变量可以实现多次初始化，而静态变量只能初始化一次。

1. 静态方法和实例方法有何不同？

静态方法和实例方法的区别主要体现在两个方面：

在外部调用静态方法时，可以使用"类名.方法名"的方式，也可以使用"对象名.方法名"的方式。而实例方法只有后面这种方式。也就是说，调用静态方法可以无需创建对象。

静态方法在访问本类的成员时，只允许访问静态成员（即静态成员变量和静态方法），而不允许访问实例成员变量和实例方法；实例方法则无此限制。

1. 在一个静态方法内调用一个非静态成员为什么是非法的？

由于静态方法可以不通过对象进行调用，因此在静态方法里，不能调用其他非静态变量，也不可以访问非静态变量成员。

1. 什么是方法的返回值？返回值的作用是什么？

方法的返回值是指我们获取到的某个方法体中的代码执行后产生的结果

返回值的作用:接收出结果，使得它可以用于其他的操作！

1. 什么是内部类？

将一个类A 定义在另一个类B里面, 里面的那个类A就称为内部类, 外面的那个类B就称为外部类

1. 内部类的分类有哪些

静态内部类：定义在类中方法外的静态类

成员内部类：定义在类中方法外的类

局部内部类：在局部位置定义的类被称为局部内部类

匿名内部类： 匿名内部类是内部类的简化写法,它的本质是一个带具体实现的父类或者父类接口的 匿名的 子类对象

1. 内部类的优点

一个内部类对象可以访问创建它的外部类对象的内容，包括私有数据！  
内部类不为同一包的其他类所见，具有很好的封装性；  
内部类有效实现了“多重继承”，优化 java 单继承的缺陷。  
匿名内部类可以很方便的定义回调。

1. 内部类有哪些应用场景

场景一：当某个类除了它的外部类，不再被其他的类使用时。我们说这个内部类依附于它的外部类而存在，可能的原因有：1、不可能为其他的类使用；2、出于某种原因，不能被其他类引用，可能会引起错误。等等。这个场景是我们使用内部类比较多的一个场景。（内部类可以看成代码隐藏机制）

场景二：当我们希望一个类必须继承多个抽象或者具体的类时，就只能用内部类来实现多重继承

1. 构造器（constructor）是否可被重写（override）

构造器(构造方法)Constructor 不能被继承，因此不能重写 Override，但可以被重载 Overload（不同参数即可）。

每一个类必须有自己的构造函数，在创建对象时自动调用，如果添加有参构造函数后，默认无参构造函数则被覆盖。子类不会覆盖父类的构造函数，但是在创建子类对象的时候，会自动调用父类构造函数。

1. 重载（Overload）和重写（Override）的区别。

方法重写：在不同的类中，指子类和父类的关系，子类重写其父类的方法，要求方法名、参数类型、参数个数必须相同，返回值类型相同 访问权限不能降低

方法重载：方法名相同 参数（数量 顺序 类型）不同与返回值类型无关

1. 重载的方法能否根据返回类型进行区分？

编译器无法在返回结果之前知道你的返回类型，所以不能通过返回类型来判断你要调用哪个函数，因此不能根据方法的返回类型来区分函数的重载。

1. == 和 equals 的区别是什么

==是判断两个变量或实例是不是指向同一个内存空间，equals是判断两个变量或实例所指向的内存空间的值是不是相同

如果对象重写了equals()方法，比较两个对象的内容是否相等；如果没有重写，比较两个对象的地址是否相同，价于“==”。

1. hashCode 与 equals (重要)

hashCode()返回该对象的哈希码值；equals()返回两个对象是否相等。

HashCode 用于在散列的存储结构中确定对象的存储地址。

如果两个对象equals()相等，那么两个对象的hashCode()方法返回的结果也必然相等。

如果两个对象的 hashCode()相同，则 equals()却不一定相等。

如果重写equals()方法，必须重写hashCode()方法，以保证equals方法相等时两个对象hashcode返回相同的值。

1. 对象的相等与指向他们的引用相等，两者有什么不同？

首先，对象在内存堆上，引用在内存栈上  
对象相等表示2个是同一对象，即同一内存，而引用相等，只是2个引用的地址值相等，都指向的堆上对象的地址

1. 当一个对象被当作参数传递到一个方法后，此方法可改变这个对象 的属性，并可返回变化后的结果，那么这里到底是值传递还是引用传递

是值传递。Java编程语言只有值传递参数。当一个对象实例作为一个参数被传递到方法中时，参数的值就是该对象的引用一个副本。指向同一个对象,对象的内容可以在被调用的方法中改变，但对象的引用(不是引用的副本)是永远不会改变的。

1. 为什么 Java 中只有值传递

java只有值传递，因为Java的引用类型传递的也是值。

1. 值传递和引用传递有什么区别

值传递： (形式参数类型是基本数据类型和String)：是指在调用函数时将实际参数复制一份传递到函数中，这样在函数中如果对参数进行修改，将不会影响到实际参数。

引用传递：

(形式参数类型是引用数据类型参数除去String)：也称为传地址。方法调用时，实际参数是对象(或数组)，这时实际参数与形式参数指向同一个地址，在方法执行中，对形式参数的操作实际上就是对实际参数的操作，这个结果在方法结束后被保留了下来，所以方法执行中形式参数的改变将会影响实际参数。

（1）在Java里面，当调用方法时，如果传入的数值为基本数据类型（包含String类型），形式参数的改变对实际参数不影响，就是值传递。

（2）在Java里面，当调用方法时，如果传入的数值为引用数据类型（String类型除外），形式参数的改变对实际参数有影响，就是引用传递。

1. JDK 中常用的包有哪些

java.lang： 这个是系统的基础i类，比如String等都是这里面的，这个包是唯一一个可以不用引入(import)就可以使用的包。

java.io： 这里面是所有输入输出有关的类，比如文件操作等。

java.nio：为了完善io包中的功能，提高io包中性能而写的一个新包

java.net: 这里面是与网络有关的类，比如URL，URLConnection等。

java.util： 这个是系统辅助类，特别是集合类Collection，List，Map等。java.sql： 这个是数据库操作的类，Connection， Statement，ResultSet等。javax.servlet：这个是JSP，Servlet等使用到的类。

1. import java 和 javax 有什么区别

刚开始的时候 JavaAPI 所必需的包是 java 开头的包，javax 当时只是扩展 API 包来说使用。然而随着时间的推移，javax 逐渐的扩展成为 Java API 的组成部分。但是，将扩展从 javax 包移动到 java 包将是太麻烦了，最终会破坏一堆现有的代码。因此，最终决定 javax 包将成为标准API的一部分。