**Java基础知识复习**

**一、类和对象：**

1. 面向对象的特点：

答：1、面向对象就是一种常见的思想。符合人们的思考习惯。

2、面向对象的出现，将复杂的问题简单化。

3、面向对象的出现，让曾经在过程中的执行者，变成了对象中的指挥者。

1. Java类继承，多态

覆盖注意事项：

1，子类方法覆盖父类方法时，子类权限必须要大于等于父类的权限。

2，静态只能覆盖静态，或被静态覆盖。

多态的好处：

提高了代码的扩展性，前期定义的代码可以使用后期的内容。

多态的弊端：

前期定义的内容不能使用(调用)后期子类的特有内容。

多态的前提：

1，必须有关系，继承，实现。

2，要有覆盖。

1、向上转型发生在编译期间，向下转型发生在运行时态。

2、变量没有覆盖的概念，父类不能引用子类中同名的变量。

3、父类只能引用子类中同名的方法

1. public,protected,private…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 类内部 | 本包 | 子类 | 外部包 |
| public | √ | √ | √ | √ |
| protected | √ | √ | √ | × |
| default | √ | √ | × | × |
| private | √ | × | × | × |

default修饰的变量对本包中其他类可见，对其他包中类不可见。

1. 抽象类与接口

抽象类和接口的异同点：

相同点：

都是不断向上抽取而来的。

不同点：

1，抽象类需要被继承，而且只能单继承。

接口需要被实现，而且可以多实现。

2，抽象类中可以定义抽象方法和非抽象方法，子类继承后，可以直接使用非抽象方法。

接口中只能定义抽象方法，必须由子类去实现。

3，抽象类的继承，是is a关系，在定义该体系的基本共性内容。

接口的实现是 like a 关系，在定义体系额外功能。

1. Object对象

equals与 == 的区别：

答：对于基本类型，== 操作符比较是变量的值。对于非基本类型，== 操作符比较的是变量在内存中的地址。Object中的equals效果等同 ==，其子类可覆盖改方法，比如String中的equals比较的是String中字符串的值，而非地址

1. Java内存模型

对于基本类型，Java使用传值方式。对于对象，Java使用传引用方式。

1. Class类型
2. 成员变量和静态变量的区别

1，两个变量的生命周期不同。

成员变量随着对象的创建而存在，随着对象的被回收而释放。

静态变量随着类的加载而存在，随着类的消失而消失。

2，调用方式不同。

成员变量只能被对象调用。

静态变量可以被对象调用，还可以被类名调用。

3，别名不同。

成员变量也称为实例变量。

静态变量称为类变量。

4，数据存储位置不同。

成员变量数据存储在堆内存的对象中，所以也叫对象的特有数据.

静态变量数据存储在方法区(共享数据区)的静态区，所以也叫对象的共享数据.

静态使用的注意事项：

1，静态方法只能访问静态成员。(非静态既可以访问静态，又可以访问非静态)

2，静态方法中不可以使用this或者super关键字。

3，主函数是静态的。

1. 静态什么时候用

1、静态变量。

当分析对象中所具备的成员变量的值都是相同的，这时这个成员就可以被静态修饰。只要数据在对象中都是不同的，就是对象的特有数据，必须存储在对象中，是非静态的。如果是相同的数据，对象不需要做修改，只需要使用即可，不需要存储在对象中，定义成静态的。

2、静态函数。

函数是否用静态修饰，就参考一点，就是该函数功能是否有访问到对象中的特有数据。简单点说，从源代码看，该功能是否需要访问非静态的成员变量，如果需要，该功能就是非静态的。如果不需要，就可以将该功能定义成静态的。当然，也可以定义成非静态，但是非静态需要被对象调用，而仅创建对象调用非静态的没有访问特有数据的方法，该对象的创建是没有意义。

1. 异常机制