JAVA面向对象总览

- 什么是面向对象

  把数据及对数据的操作方法放在一起，作为一个相互依存的整体——对象
- 为什么要用面向对象

  面向对象解决了系统的可维护性，可扩展性，可重用性
- 面相对象的三大特征

  封装 继承 多态

封装

- 封装的含义

```
1.把对象的状态和行为看成一个统一的整体，将二者存放在一个独立的模块中(类)；
 2."信息隐藏"，把不需要让外界知道的信息隐藏起来,尽可能隐藏对象功能实现细节,字段;
```

- 封装关键字

```
  public公共的
  private私有的
  protected受保护的
```

- this关键字

```
 1.This关键字代表当前对象
    this.属性  操作当前对象的属性
    this.方法  调用当前对象的方法2.封装对象属性的时候，经常会使用this关键字
```

- 封装实例

```
//创建一个类 将属性私有化 编写get set方法
class Class {
    private String name
    private int persones
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getPersones() {
        return persones;
    }
    public void setPersones(int persones) {
        this.persones = persones;
    }

}
```

```
//在主方法中调用Class类中的set get 方法
public class Demo {
    public static void main(String args []){
        Class cl = new Class();
        cl.setPersones(23);
        cl.setName("java");
        System.out.println(cl.getName()+"班，"+"共有"+c1.getPersone
s()+"人。");
    }
}
```

- 输出结果：java班，共有23人。

继承

- 继承概念

```
  继承是类与类的一种关系，是一种is a 的关系
  注：java中的继承是单继承
```

- 继承的好处

```
1.子类直接拥有了父类所有的属性和方法（父类私有属性不可继承 private）
2.实现代码的复用
```

- final关键字

```
final修饰类   该类不允许被继承
final修饰方法  该方法不允许被覆写
final修饰属性   该类的属性不会进行隐式的初始化或在构造方法中赋值
final修饰变量  该变量只能赋值一次即变为常量
```

- super关键字

```
在对象的内部使用，可以代表父类对象。
super.age 访问父类属性
super.eat() 访问父类方法
super必须写在第一行
```

- super的应用

```
子类的构造过程中必须调用其父类的构造方法
如果子类的构造方法中没有显示调用父类的构造方法，则系统默认调用父类无参的构造方法
如果显示的调用构造方法，必须在子类的构造方法的第一行
如果子类构造方法中既没有显示调用父类的构造方法，而父类有没有无参的构造方法，则编译出错
                                （当我们定义一个有参的构造方法，系统不会为我们定义无参的构造方法）
```

多态

- 多态的含义

```
    对象的多种形态
    父类的引用可以指向本类对象
    父类的引用可以指向子类对象
```

- 抽象类&&接口

```
abstract     允许存在普通方法
interface
```

- 继承&&多态实例

```
class Dog{
    String name;
    public void eat(){
        System.out.println();
    }
}
class Tdi extends Dog{
    public void eat(){
        System.out.println("泰迪");
    }
}
class Erha extends Dog{
    public void eat(){
        System.out.println("二哈");
    }
}
public class Main{
    public static void main(String args []){
        Dog dog = new Tdi();
        dog.eat();
    }
}
```

输出结果：泰迪