## **11.15\_常见对象(Object类的概述)(了解)**

A:Object类概述

类层次结构的根类，也称作是超类

所有类都直接或者间接的继承自该类

B:构造方法

public Object()

回想面向对象中为什么说：

子类的构造方法默认访问的是父类的无参构造方法,**super()**

## **11.16\_常见对象(Object类的hashCode()方法)**

A:案例演示

public int hashCode()

a:返回该对象的哈希码值。**默认情况下，该方法会根据对象的地址来计算。**

**地址就是十进制的数,而且是随机进行分配的,.但是一般自定义的类是重写hashcode()的让其与自身的属性相关联创造出有意义的地址值,也就是为了尽量使对象的hashcode不一样.**

b:**不同对象的，hashCode()一般来说不会相同（也有相同的）。但是，同一个对象的hashCode()值肯定相同。**

快捷键就是 **ctrl + alt +　上下键**

---------------------------------------------------------------------------------------------------------------------------

Object h = new Object(); //利用Object超类进行创建对象

int g = h.hashCode();

System.out.println(g); //返回值是int类型，接收并打印地址值

Person a = new Person("张三",23);

Person b = new Person("张三",23);

System.out.println(a); //两个对象的地址值是不一样的。

System.out.println(b);

---------------------------------------------------------------------------------------------------------------------------

## **11.17\_常见对象(Object类的getClass()方法)(在反射的时候掌握)（获取类名）**

A:案例演示

public final Class getClass()

a:返回此 Object 的运行时类。

b:可以通过Class类中的一个方法，获取对象的真实类的全名称。

public String getName()

---------------------------------------------------------------------------------------------------------------------------

import com.heima.Person;

public class Student {

public static void main(String[] args) {

Person s = new Person("张三",23);

Class clazz = s.getClass(); //拿到对象的的字节码文件，就是编译之后的文件

String name = clazz.getName(); //调用getname方法获取名称

System.out.println(name); //打印名称，其实就是对象s的类名

**System.out.println(s.getClass().getName()); //两步和做一部进行也是可以的。最后打印的是person的全类名就是包名加类名。s属于person类。**

---------------------------------------------------------------------------------------------------------------------------

## **11.18\_常见对象(Object类的toString()方法)(掌握)（获取成员属性ctrl + alt + S--->S）**

A:案例演示

public String toString()

a:返回该对象的字符串表示。

b:它的值等于：

public Stirng toString() {

return getClass().getName() + "@" + Integer.toHexString(hashCode())

}

上边就是toString()方法的源码. 就是类名 @ 十六进制的地址值

com.heima.bean.Student@bcda2d,这就是不重写object类的toString()方法的打印结果

c:由于默认情况下的数据对我们来说没有意义，一般建议重写该方法。所有子类都包含有这个类，而且每一次都是自动启动的，但是地址值又没有意义所以干脆直接改写挺好的啊，省空间！,**软件重写了之后toString方法的作用:可以更方便的显示属性值(当然我们也可以自己改,开发中都是用软件重写的)**。比如说对象就是显示的属性name和age什么的,但是我们为什么不去采用的getxxx方法呢?**getXxx方法是为了获取值，可以显示也可以赋值，或者其他操作。打印的时候自动调用了toString() 方法,所以打印s和打印s.toString()结果是一样的.**

B:最终版

自动生成的时候就是直接选择name 和 age 即可，不用选择其他的项目。

---------------------------------------------------------------------------------------------------------------------------

public class Demo3\_ToString {

public static void main(String[] args) {

Student s = new Student("张三", 23);

// String str = s.toString();

// System.out.println(str);

System.out.println(s.toString());

System.out.println(s);

System.out.println("我的姓名是：" + s.getName() + ",我的年龄是："+ s.getAge());

}

}

---------------------------------------------------------------------------------------------------------------------------

## **11.19\_常见对象(Object类的equals()方法)(掌握)（ctrl + alt + s—>v）**

A:案例演示

a:指示其他某个对象是否与此对象“相等”。

b:默认情况下比较的是对象的引用是否相同(object里边比较的是hashcode值,而且object中hashcode由哈说表提供的)。

c:由于比较对象的引用没有意义(object中equals源码是this==obj,所以是地址值)，一般建议重写该方法.比较对象的属性值.

---------------------------------------------------------------------------------------------------------------------------

public boolean equals(Object obj) { //重写父类方法，后边的也是父类参数obj

Person s = (Person)obj; //将父类obj强转成为子类

return this.name.equals(s.name) && this.age == s.age ; //调用子类String的equals方法。int属于是基本数据类型只能是运算符的比较。This.name是属于String的对象，所以可以直接性的调用。

---------------------------------------------------------------------------------------------------------------------------

## 11.20\_常见对象(==号和equals方法的区别)(掌握)

==是一个比较运算符号,既可以比较基本数据类型,也可以比较引用数据类型,基本数据类型比较的是值,引用数据类型比较的是地址值

equals方法是一个方法,只能比较引用数据类型,所有的对象都会继承Object类中的方法,如果没有重写Object类中的equals方法,equals方法和==号比较引用数据类型无区别,重写后的equals方法比较的是对象中的属性（String不属于基本数据类型属于引用数据类型,String 是字符串类,String中同时改写了equals和hashcode方法,按说改写了equals方法即可,同时改写hashcode方法是为了提高哈说表的维护性,hashcode算法都不尽一样,String的和integer的就不一样.但是我们一般很少书写对象的equals方法,因为同时亦需要书写hashcode自己写一定的算法）

注意:String类在当做参数传递的时候是要当做是基本数据类型的,因为是在常量池中的,而且是不可改变的.

11.21\_day11总结

把今天的知识点总结一遍。