对象转型、Object类、多态

回顾

今天任务

1. Object 类
2. 多态
3. 向上转型
4. 向下转型

教学目标

1. 掌握Object类中的方法
2. 掌握多态
3. 向上转型
4. 向上转型

第一节： Object类

> 类Object是类层次结构的根类。每个类都使用Object作为超类。

1.1 equals()方法

> 指示其他某个对象是否与此对象“相等”。

```
String s1 = new String("abc");
String s2 = new String("abc");
System.out.println(s1==s2);//new表示在堆中开辟新的空间存储数据，双等号比较的是new出来的两块对内存
System.out.println(s1.equals(s2));//String类中重写了equals方法，比较的是两个字符串内容是否相等
```

1.2 hashCode()方法

> 返回该对象的哈希码值。每一个对象的哈希码值唯一

```
Object obj1 = new Object();
Object obj2 = new Object();
Object obj3 = obj2;
//obj2与obj3两个对象的哈希码值一样
System.out.println(obj1.hashCode());
System.out.println(obj2.hashCode());
System.out.println(obj3.hashCode());
```

1.3 getClass()方法

> 返回此Object的运行时类。通过某个对象通过反射获取类。

```
    Object obj = new Object();
    Class cls = obj.getClass();//通过反射获取到了Object类
```

1.4toString()方法

> 返回该对象的字符串表示。可以在自定义类中重写toString方法，以实现是对象转成指定格式的字符串。

```
public class Person{
    String name;
    public Person(String name){
        this.name = name;
    }
    //重写toString方法
    public String toString(){
        return "name="+name;
    }
}
public class DemoPerson{
    public static void main(String[]args){
        Person p = new Person("张三");
        System.out.println(p);//输出结果为：name=张三
    }
}
```

第二节： 多态

2.1 什么是多态

> 一个事物有多种表现形态
>
> 对象的多种形态，方法的多样性。

2.2 向上转型，向下转型

> 向上转型：将子类的引用赋值给父类引用，自动转换
>
> ```
> String str = "abc";
> Object obj = str;
> ```
>
> 向下转型：将父类的引用赋值给子类引用，强制转换

```
Object obj = new String("abc");
String str = (String)obj;
```

2.3 instanceof

> 对象向下转型时，存在一个问题：
>
> 若一个父类A的引用a，指向一个子类B的实例，将a赋值给另一个子类C引用时，会抛出java.lang.ClassCastException异常；
>
> 抛出异常后程序将不能继续向下执行，为了避免这个异常的抛出，我们可以通过instanceof关键字判断引用指向的实例是否可以进行强制转换成某个子类对象

> 已知Animal类有两个子类Dog和Cat：

```
public class Demo{
    public static void main(String[]args){
        Animal a = new Dog();
        check(a);
    }
    //设计一个方法，判断一个动物是猫还是狗
    public static void check(Animal a){
        if(a instanceof Dog){
            System.out.println("狗");
        }else if(a instanceof Cat){
            System.out.println("猫");
        }
    }
}
```

2.4 多态的体现

> 对象的多态：
>
> 父类引用指向子类实例
>
> 接口引用指向实现类实例
>
> 方法的多样性：方法名相同，执行不同
>
> 方法的重写：子类中对父类继承来的方法进行完善
>
> 方法的重载：方法名相同参数列表不同

课前默写

> 设计一个图书类Book，包括私有属性：编号、书名、价格，实现图书自动编号 （图书自动编号）

设计一个工具类Tools，工具类中设计以下方法： 1）在一个图书数组中按编号查找图书，并将图书信息输出， 分析： 返回值：可以不要返回值 参数：需要，两个，Book数组、编号 2）在一个图书数组中按价格查找图书，并将查找到的所有图书返回（注意考虑相同价格的图书，返回一个数组） 分析： 返回值：有，一个新数组（Book），是查询到指定价格的图书 参数：需要，两个，Book数组，价格

在主方法中创建一个图书数组，调用工具类中的按编号查找图书，再调用按价格查找图书，再将所有查找到的图书信息输出

图书信息的输出格式：编号是xx，书名xxx，价格是xx

作业

电话本

实现功能: 请输入要选择的功能:

1. 新增一个联系人
2. 查找指定的联系人
3. 查找所有的联系人
4. 退出系统

功能要求: 1、新增一个联系人，联系人的姓名，性别，年龄，手机号，身份证号由用户从控制台录入 2、按照姓名查找符合条件的所有的联系人(有可能有联系人同名情况，如果同名，全部列出) 3、查询所有的联系人 4、该电话本只能存最多10个联系人 涉及类: 电话本 联系人

面试题

```
class A{
    public String show(D obj){
           return ("A and D");
    }
    public String show(A obj){
           return ("A and A");
    }
}
class B extends A{
    public String show(B obj){
           return ("B and B");
    }
    public String show(A obj){
           return ("B and A");
    }
}
class C extends B{}
class D extends B{}
```

问题：以下输出结果是什么？

A a1 = new A(); A a2 = new B(); B b = new B(); C c = new C(); D d = new D();
System.out.println(a1.show(b)); ① System.out.println(a1.show(c)); ② System.out.println(a1.show(d)); ③
System.out.println(a2.show(b)); ④ System.out.println(a2.show(c)); ⑤ System.out.println(a2.show(d)); ⑥
System.out.println(b.show(b)); ⑦ System.out.println(b.show(c)); ⑧ System.out.println(b.show(d)); ⑨