1 引用类型的转换

- 上溯造型(即向上转型):子类对象转成父类类型.也叫作自动类型转换，必须由继承或者实现关系。需要注意的是，父类引用指向子类对象，会丢失子类新扩展的属性和方法，只能调用父类和子类共有的属性和方法。

```
package com.highcom;

public class Animal {
    private String name;
    public Animal() {
        super();
    }
    public Animal(String name) {
        super();
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

```java
package com.highcom;

public class Cat extends Animal{

    public Cat() {
        super();
        // TODO Auto-generated constructor
stub
    }

    public Cat(String name) {
        super(name);
        // TODO Auto-generated constructor
stub
    }
}
```

```java
public interface Consumer {

    public void pay();
}
```

```java
package com.highcom;

public class Employee implements Consumer{

    private String name;
    private int age;
    private String sex;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getSex() {
        return sex;
    }

    public void setSex(String sex) {
        this.sex = sex;
    }

    public Employee() {
        super();
    }

    public Employee(String name, int age,
String sex) {
        super();
        this.name = name;
        this.age = age;
        this.sex = sex;
    }

    @Override
    public void pay() {
        System.out.println(name+",这个员工使用现
金进行消费！");
    }
}
```

- 下溯造型(即向下转型):父类对象转成子类类型.也叫作强制类型转换,引用类型中的强制类型转换，有个前提，必须向上转型后，才能向下转型.

  向下转型的语法:

  Animal a = new Cat();

  Cat c = (Cat)a;

```
package com.highcom;
```

```
public class Worker extends Employee
implements Consumer{

    private double addressAllowance;

    public double getAddressAllowance() {
        return addressAllowance;
    }

    public void setAddressAllowance(double
addressAllowance) {
        this.addressAllowance =
addressAllowance;
    }

    @Override
    public void pay() {
        System.out.println("当前的工人使用微信进
行消费!");
    }
}
```

```
package com.highcom;

public class TestWorker {

    public static void main(String[] args) {
        //创建工人对象
        Worker w = new Worker();
        w.setName("洛奇");
        w.setAge(30);
        w.setSex("男");
        w.setAddressAllowance(2000);

        System.out.println("工人基本信息：");

System.out.println(w.getName()+","+w.getAge()
+","+w.getSex()+",住房补
助："+w.getAddressAllowance());

        //向上转型
        Employee emp = w;
        //向下转型
        Worker w1 = (Worker)emp;

//System.out.println(w1.getAddressAllowance()
);
//      w1.pay();
        //w1是什么类型?
        //可以使用 instanceof 进行对象类型的判断
        // 对象名 instanceof 类型名
        System.out.println(w1 instanceof
Employee);
        System.out.println(w1 instanceof
Worker);
```

```
        System.out.println(w1 instanceof Consumer);//
    }
}
```