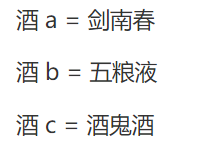
面向对象编程三大特性

封装，继承，多态

多态



不知道何种二维码，只有在运行的时候才知道



我们只知道他是酒，喝了才知道是啥  
这就是多态

好处：增加代码可读性，可以统一管理，拓展

向上转型，我们declare的是酒，但是我们可以通过给它定义子类，添加不同的功能,剑南春被**向上转型**成了酒

坏处在于，它向上转型以后，如果有什么method是剑南春独有而酒没有的，那么注定无法使用这个method,除非使用instance of

public class Wine {

public void fun1(){

System.out.println("Wine 的Fun.....");

fun2();

}

public void fun2(){

System.out.println("Wine 的Fun2...");

}

}

public class JNC extends Wine{

/\*\*

\* @desc 子类重载父类方法

\* 父类中不存在该方法，向上转型后，父类是不能引用该方法的

\* @param a

\* @return void

\*/

public void fun1(String a){

System.out.println("JNC 的 Fun1...");

fun2();

}

/\*\*

\* 子类重写父类方法

\* 指向子类的父类引用调用fun2时，必定是调用该方法

\*/

public void fun2(){

System.out.println("JNC 的Fun2...");

}

}

public class Test {

public static void main(String[] args) {

Wine a = new JNC();

a.fun1();

}

}

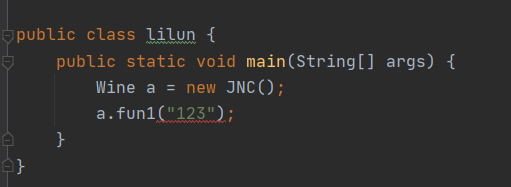
-------------------------------------------------

Output:

Wine 的Fun.....

JNC 的Fun2...

优先级：子类有这个function就用子类的，没有·就用父类的，但是如果是子类有这个function而父类没有，那么不能call这个function，因为我们没运行不知道他具体是哪种



这里fun1(String a)是子类JNC里的，我们没运行不知道他是剑南春，所以不能用

         继承：在多态中必须存在有继承关系的子类和父类。

         重写：子类对父类中某些方法进行重新定义，在调用这些方法时就会调用子类的方法。

         向上转型：在多态中需要将子类的引用赋给父类对象，只有这样该引用才能够具备技能调用父类的方法和子类的方法。

         只有满足了上述三个条件，我们才能够在同一个继承结构中使用统一的逻辑实现代码处理不同的对象，从而达到执行不同的行为。