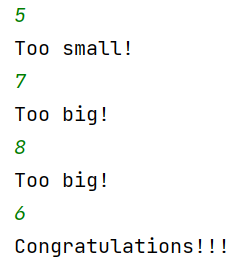


同时创建package与类：package.class

猜数字：

public void guess()  
{  
 System.*out*.println("guess from 0 to 100:");  
 Random r = new Random();  
 int num=r.nextInt(101);  
 Scanner s = new Scanner(System.*in*);  
 int input=s.nextInt();  
  
 while(input!=num)  
 {  
 if(input>num)  
 {  
 System.*out*.println("Too big!");  
 input=s.nextInt();  
 continue;  
 }  
 else if(input<num)  
 {  
 System.*out*.println("Too small!");  
 input=s.nextInt();  
 continue;  
 }  
 }  
 System.*out*.println("Congratulations!!!");  
}



Java中==比较的是字符串的地址，比较值：s1.equals(s2);

@Override：重写，子类重写父类方法 @代表注解

注释：给人看的，注解：给系统、程序看的

Arraylist: add remove size get

接口：高度抽象类，只有类实现了接口中的所有方法才能使用接口的变量

构造类：

Public student（string name）

{

This.name=name;

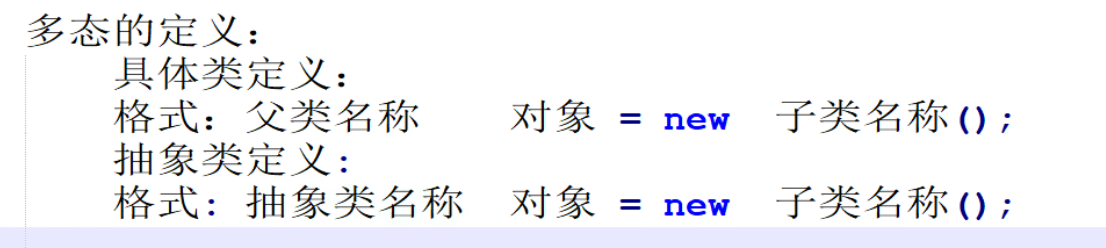
}

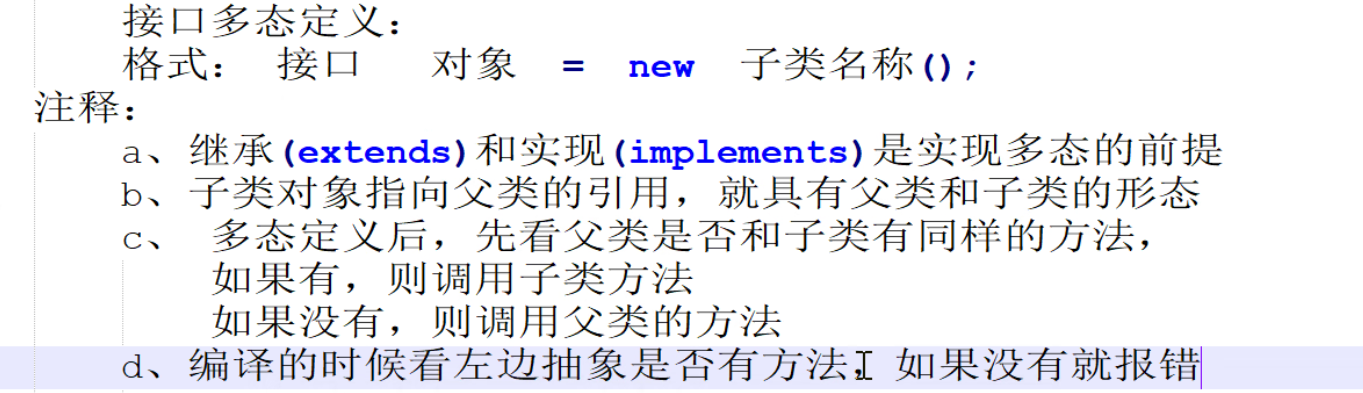
子类函数中调用父类函数：super.xxx();

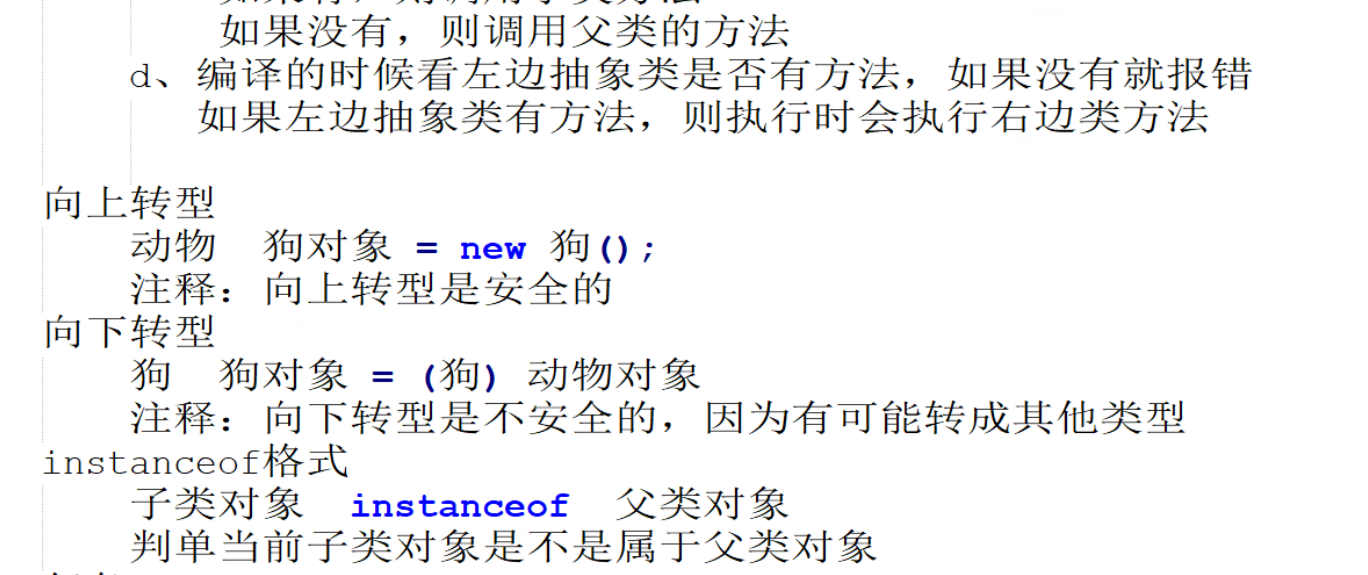
在java中，super表示[超类](https://so.csdn.net/so/search?q=%E8%B6%85%E7%B1%BB&spm=1001.2101.3001.7020)（就是我们俗称的父类），当子类需要引用父类的字段时，我们就可以使用super.FieldName.

Alt+enter：快速实现抽象类方法

Static函数可以直接在类中调用，若不是static的话必须由某个对象调用







Usb问题：

Usb接口：

package Users;  
  
public interface Usb {  
 void start();  
 void run();  
 void stop();  
}

正品Mouse实现了接口：

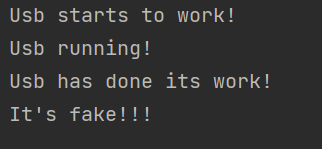
package Users;  
  
public class MouseUsb implements Usb{  
 public void start()  
 {  
 System.*out*.println("Usb starts to work!");  
 }  
 public void run()  
 {  
 System.*out*.println("Usb running!");  
 }  
 public void stop()  
 {  
 System.*out*.println("Usb has done its work!");  
 }  
}

山寨micro无法实现接口，故不作实现

Demo：

package Users;  
  
public class Demo {  
 public static void main(String[] args) {  
 *click*(new MouseUsb());  
 *click*(new MicroUsb());  
 }  
 public static void click(Object Usb) {  
 if (Usb instanceof MouseUsb) {  
 ((MouseUsb) Usb).start();  
 ((MouseUsb) Usb).run();  
 ((MouseUsb) Usb).stop();  
 } else  
 {  
 System.*out*.println("It's fake!!!");  
 }  
 }  
}

运行结果如下



发红包问题：

我设立的情景是由输入者添加群主、群成员，并控制群主发多少红包、谁来抢，直到群主没钱为止。（只有群主可以发红包）

User大类：包括群主、群成员

package RedEnvelope;  
  
import java.util.ArrayList;  
  
public class User {  
 int balance;  
 String name;  
 String id;  
 public void show()  
 {  
 System.*out*.println(this.name+" still got:"+this.balance);  
 }  
  
 public ArrayList<Integer> receive(ArrayList<Integer> l)  
 {  
 this.balance+=l.get(0);  
 l.remove(0);  
 return l;  
 }  
}

member类：只需继承即可，无特殊功能

package RedEnvelope;  
  
import java.util.ArrayList;  
  
public class Member extends User {  
  
  
}

manager类：在能抢红包的基础上需要能发红包

package RedEnvelope;  
  
import java.util.ArrayList;  
import java.util.Random;  
import java.util.Scanner;  
  
public class Manager extends User {  
  
  
  
 public ArrayList<Integer> send(int num,int pnum)  
 {  
 Random r = new Random();  
 this.balance-=num;  
 ArrayList<Integer> elist = new ArrayList<>();  
 for(int i=0;i<pnum-1;i++)  
 {  
 int money=r.nextInt(num);  
 elist.add(money);  
 num-=money;  
 }  
 elist.add(num);  
 return elist;  
 }

group类：输入者创建群聊

package RedEnvelope;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Group extends User{  
 public static void main(String[] args) {  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Who is the Manager?");  
 System.*out*.println("Please type in his name,id,and balance");  
 Manager ma = new Manager();  
 ma.name=s.next();  
 ma.id=s.next();  
 ma.balance=s.nextInt();  
 System.*out*.println("Who are the Members?");  
 System.*out*.println("Please type in their names,ids,and balances (Type 'exit' to stop)");  
 ArrayList<Member> memlist = new ArrayList<>();  
 String name=null;  
 while(!(name=s.next()).equals("exit"))  
 {  
 Member me = new Member();  
 me.name=name;  
 me.id=s.next();  
 me.balance=s.nextInt();  
 memlist.add(me);  
 }  
 //群主、群成员选择完毕，建群成功！  
 System.*out*.println("Start to send envelopes!");  
 while(ma.balance>0)  
 {  
 System.*out*.println("How much money & How many envelopes do manager wants to send?");  
 System.*out*.println("Money:");  
 int num=s.nextInt();  
 System.*out*.println("Envelopes:");  
 int pnum=s.nextInt();  
 ArrayList<Integer> envelope = ma.send(num,pnum);  
 System.*out*.println("Who receives the envelope? (Type 'exit' to stop)");  
 int times=0;  
 String id=null;  
 while(!(id=s.next()).equals("exit"))  
 {  
 if(times==pnum) //红包已被抢完，循环停止  
 {  
 System.*out*.println("There is no envelope left!");  
 break;  
 }  
 else  
 {  
 times++;  
 if(ma.id.equals(id)) //群主抢红包  
 {  
 envelope=ma.receive(envelope);  
 ma.show();  
 }  
 else  
 {  
 int index=0;  
 for(int i=0;i<memlist.size();i++)  
 {  
 if(memlist.get(i).id.equals(id))  
 {  
 index=i;  
 break;  
 }  
 }  
 envelope=memlist.get(index).receive(envelope);  
 memlist.get(index).show();  
 }  
 }  
 }  
 if(times<pnum) //红包没被抢光但抢红包的过程已结束，则退还钱给群主  
 {  
 System.*out*.println("There are still some envelopes left");  
 for(int i=0;i<envelope.size();i++)  
 {  
 ma.balance+= envelope.get(i);  
 }  
 envelope.clear();  
 System.*out*.println("The money has been sent back to Manager");  
 ma.show();  
 }  
 }  
  
 System.*out*.println("The Manager got no money to send!");  
 }  
}

//测试结果如下

