一种语法是的新东西Lambda，lambda是一个匿名函数，使用lambda可以更灵活

方法引用

Integer ：： compare

-> : lambda操作符 或者箭头操作符

->左边 ：lambda形参列表，（其实就是抽象方法的形参列表）

->右边 ：lambda体， （其实是重写的抽象函数的方法体）

lambda表达式的本质 ：接口的实例，接口只能有一个方法，是函数式接口

lambda表达式使用 （分为六种情况）

1，无形参 ，无返回值

Runnable r2 = （）-> System.out.printf("....")

2.lambda 需要一个参数，但是没有返回值

Consumer<String> con = (String s) ->System.out.printf(s);

3.数据类型可以省略，因为可以由编译器推断得出，称为“类型推断”

Consumer<String> con = (s) ->System.out.printf(s);

因为Consumer<String> 已经定过了String类型，可以推断出s为String类型

4.Lambda 若只需要一个参数时，参数的小括号可以省略

Consumer<String> con = s ->System.out.printf(s);

5.lambda如果需要两个或以上的参数，并且执行多条执行语句，并且可以有返回值

Comparator<Integer> integerComparator = (o1,o2)->{

System.***out***.println(o1);

**return** (o1 > o2) ? o1 : o2;

};

6.当Lambda只有一条语句，return与大括号若有，都可以省略，必须同时都省略

Comparator<Integer> integerComparator = (o1,o2)-> (o1 > o2) ? o1 : o2;

总结：lambda形参只有一个参数，括号可以省略，如果lambda体只有一条执行语句，（可能是return语句），就可以省略大括号

上界<? extends T>不能往里存，只能往外取

下界<? super T>不影响往里存，但往外取只能放在Object对象里