**作业**

任务一：

定义一个泛型类盒子Box<T>,属性width和height,

调用验证支持Intger,Double,Streing类型。

知识点：泛型

package day14.homeWork;

class Box<T>{

private T width;

private T height;

public T getWidth() {

return width;

}

public void setWidth(T width) {

this.width = width;

}

public T getHeight() {

return height;

}

public void setHeight(T height) {

this.height = height;

}

}

public class ZuoYe1 {

public static void main(String[] args) {

Box<Integer> z1 = new Box<Integer>();

z1.setHeight(45);

z1.setWidth(40);

Box<Double> z2 = new Box<Double>();

z2.setHeight(45.2);

z2.setWidth(40.2);

Box<String> z3 = new Box<String>();

z3.setHeight("5m");

z3.setWidth("3m");

}

}

任务二：

定义一个季节枚举类，枚举成员只有春、夏、秋、冬。

实现控制台输入季节，

符合春天，显示“春暖花开”；

符合夏天，显示“夏日炎炎”；

符合秋天，显示“秋高气爽”；

符合冬天，显示“冬日雪飘”。

package day14.homeWork;

import java.util.Scanner;

enum Season{

SPRING("春天","春暖花开"),SUMMER ("夏天","夏日炎炎"),

FALL("秋天","秋高气爽"),WINTER("冬天","冬日雪飘");

private String msg;

private String season;

private Season(String season,String msg) {

this.msg = msg;

this.season = season;

}

public String getMsg() {

return msg;

}

public void setMsg(String msg) {

this.msg = msg;

}

public String getSeason() {

return season;

}

public void setSeason(String season) {

this.season = season;

}

@Override

public String toString() {

// TODO Auto-generated method stub

return season+" "+ msg;

}

}

public class ZuoYe2 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("请输入季节：");

String season = sc.next();

for (Season s : Season.values()) {

if(s.getSeason().equals(season)) {

System.out.println(s);

break;

}

}

/\*Season season = Season.valueOf(sc.next());

switch(season) {

case SPRING:

System.out.println(SPRING);

break;

case SUMMER:

System.out.println(SUMMER);

break;

case FALL:

System.out.println(FALL);

break;

case WINTER:

System.out.println(WINTER);

break;

}\*/

}

}

任务三：

定义一个会员类，会员有编号，名字，积分。

1.编写外部比较器，实现按积分降序排序。

2.编写一个内部比较器，按照编号进行升序排序。

package day14.homeWork;

import java.util.Arrays;

import java.util.Comparator;

class Member implements Comparable<Member>{

private int no;

private String name;

private int score;

public Member(int no, String name, int score) {

super();

this.no = no;

this.name = name;

this.score = score;

}

public int getNo() {

return no;

}

public void setNo(int no) {

this.no = no;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getScore() {

return score;

}

public void setScore(int score) {

this.score = score;

}

@Override

public int compareTo(Member o) {

// TODO Auto-generated method stub

return this.no - o.no;

}

@Override

public String toString() {

return "Member [no=" + no + ", name=" + name + ", score=" + score + "]";

}

}

public class ZuoYe3 {

public static void main(String[] args) {

Member [] m = new Member[3];

m[0] = new Member(2,"张三",53);

m[1] = new Member(1,"李四",89);

m[2] = new Member(3,"王五",63);

//内部比较器，按照编号进行升序排序

//Arrays.sort(m);

//外部比较器，实现按积分降序排序。

Arrays.sort(m, new Comparator<Member>() {

@Override

public int compare(Member o1, Member o2) {

// TODO Auto-generated method stub

return o2.getScore() - o1.getScore();

}

});

//外部比较器，实现按积分降序排序。lambda表达式

//Arrays.sort(m,(m1,m2)->m2.getScore()-m1.getScore());

Arrays.asList(m).forEach(System.out::println);

}

}