1. [Comparable]: 完成以下ParttimeStudent與ParttimeStudentX使產生規定功能。

[程式片段]

import java.util.Arrays;

class Student\_2 implements Comparable {

String name ; int score ;

Student\_2() { }

Student\_2(String y, int x) { score= x; name=y; }

Student\_2(Student\_2 stu) { score=stu.score; name=stu.name; }

public int compareTo(Object o) {

// 先比score，若相等，再比name

return 1;

}

public String toString() { return "("+name+","+score+")";}

}

// --- 使用繼承 ---

class ParttimeStudent extends Student\_2 {

double salary ; // key-3

ParttimeStudent() { }

// DIY: 參考main()中的需求，自行加入合適的建構式

public int compareTo(Object o) { // DIY

// 若id, name相同，則以salary決定大小

return 1 ;

}

// DIY: 自行加入toString(), 格式自行設計

}

// --- 使用聚合 ---

class ParttimeStudentX implements Comparable {

Student\_2 stu ;

double salary ; // key-3

ParttimeStudentX() { }

// DIY, 參考main()中的需求，自行加入合適的建構式

public int compareTo(Object o) {

/\* DIY here \*/

return 1 ;

}

// DIY: 自行加入toString(), 格式自行設計

}

public class HWK6\_XX\_YY\_01 {

public static void main(String[] args) {

Student\_2 s1 = new Student\_2("Mary",85) ;

Student\_2 s2 = new Student\_2("Peter",73) ;

Student\_2 s3 = new Student\_2("Mary",93) ;

Student\_2[] stus = {s1,s2,s3} ;

Arrays.sort(stus) ;

System.out.println(Arrays.asList(stus)) ;

ParttimeStudent ps1 = new ParttimeStudent(s1,35000) ;

ParttimeStudent ps2 = new ParttimeStudent(s2,25000) ;

ParttimeStudent ps3 = new ParttimeStudent(s2,29000) ;

ParttimeStudent[] pss = {ps1,ps2,ps3} ;

Arrays.sort(pss) ;

System.out.println(Arrays.asList(pss)) ;

ParttimeStudentX psx1 = new ParttimeStudentX(s1,35000) ;

ParttimeStudentX psx2 = new ParttimeStudentX(s2,25000) ;

ParttimeStudentX psx3 = new ParttimeStudentX(s2,29000) ;

ParttimeStudentX[] psxs = {psx1,psx2,psx3} ;

Arrays.sort(psxs) ;

System.out.println(Arrays.asList(psxs)) ;

}

}

1. [Comparable]: 請完成以下類別中的compareTo()，以產生規定功能。

[程式片段]

import java.util.\*;

class Product {

int a, b, c ; // a\*b\*c

Product(int \_a, int \_b, int \_c) {a=\_a; b=\_b; c=\_c;}

// compareTo(): 比較乘積的大小

public String toString() {

return "("+a+","+b+","+c+")" ;

}

}

class ProductArray {

TreeSet<Product> data =new TreeSet<Product>() ;

// 提示: data.last()最大值, data.first()最小值

void add(Product p) { data.add(p);}

// compareTo(): 以data中最大值(Product)做為比較大小的依據

public String toString() { return data.toString(); }

}

public class HWK6\_XX\_YY\_02 {

public static void main(String[] args) {

Random r = new Random(111) ;

ProductArray[] pas = new ProductArray[4] ;

// 亂數產生一個ProductArray的陣列

for (int i=0; i<pas.length; i++) {

pas[i] = new ProductArray() ;

for (int j = 0; j<10; j++)

pas[i].add(new Product(r.nextInt(101), r.nextInt(101), r.nextInt(101))) ;

System.out.println("ProductArray-"+i+"="+pas[i]) ;

}

Arrays.sort(pas) ;

// --- 印出pas[]中最大的ProductArray物件

System.out.println(pas[pas.length-1]) ;

}

}

1. [Comparator]: 請在以下程式加入一GPAComparator，使學生能依照GPA由小到大排序，並同時在main()中增加新程式碼，以印出結果。

[程式片段]

import java.util.\*;

class Student3 {

int id; String name ; double gpa ;

Student3() { }

Student3(int \_x, String \_y, double g) { id= \_x; name=\_y; gpa=g;}

public String toString() {

return "("+id+","+name+","+gpa+")";

}

}

class IDComparator implements Comparator<Student3> {

public int compare(Student3 s1, Student3 s2) {

if (s1.id>s2.id) return 1 ;

if (s1.id<s2.id) return -1 ;

return s1.name.compareTo(s2.name) ;

}

}

public class HWK6\_XX\_YY\_03 {

public static void main(String[] args) {

Student3[] studs= new Student3[10] ;

for (int i = 0 ; i<studs.length; i++)

studs[i] = new Student3((int)(Math.random()\*1000),

("Name\_"+i),(2.5+(int)(Math.random()\*5)\*0.5)) ;

System.out.println("Before Sort(): "+Arrays.asList(studs)) ;

Arrays.sort(studs, new IDComparator()) ;

System.out.println("After Sort(): "+Arrays.asList(studs)) ;

}

}