姓名： 谢宝玛

学号：1120233506

实验日期： 2025-03-14

实验一：

1. **实验内容**

实现一个名为Value 的类，其中含有一个字符类型的成员x 和双精度类型的

成员y。请重写（override）其equals( )方法，使得用equals( )比较Value 类的两

个对象的内容时，相等条件是对象的所有成员分别对应相等。

1. **实验思路**

equals( )比较Value 类的两个对象的内容时，相等条件是对象的所有成员分别对应相等

1. **实验源码**

**Value：**import java.util.Objects;  
  
public class Value {  
 private char x;  
 private double y;  
  
 public Value(char x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
  
 public char getX() {  
 return x;  
 }  
  
 public void setX(char x) {  
 this.x = x;  
 }  
  
 public double getY() {  
 return y;  
 }  
  
 public void setY(double y) {  
 this.y = y;  
 }  
  
 @Override  
 public boolean equals(Object obj) {  
 if (this == obj) return true;  
 if (obj == null || getClass() != obj.getClass()) return false;  
 Value value = (Value) obj;  
 return x == value.x && Double.compare(value.y, y) == 0;  
 }  
}

**TestValue:**

import java.util.HashSet;  
  
public class TestValue {  
 public static void main(String[] args) {  
 // 创建两个相同的 Value 对象  
 Value v1 = new Value('A', 10.5);  
 Value v2 = new Value('A', 10.5);  
 Value v3 = new Value('B', 20.0);  
  
 // 测试 equals 方法  
 System.out.println("v1 equals v2: " + v1.equals(v2)); // 应该返回 true  
 System.out.println("v1 equals v3: " + v1.equals(v3)); // 应该返回 false  
 }  
}

1. **实验心得**

注意重写方法的语法

实验二

1. **实验内容**

实现下列表格内容 ，并且分别按Name、Age（倒序）、Grade排序。内容如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Age | Grade |
| 1 | ZhangSan | 28 | 98 |
| 2 | LiSi | 21 | 100 |
| 3 | KangKang | 27 | 89 |
| 4 | LiMing | 19 | 92 |
| 5 | WangGang | 22 | 66 |
| 6 | ZhaoXin | 24 | 85 |
| 7 | LiuWei | 20 | 78 |
| 8 | BaiZhanTang | 16 | 99 |

1. **实验思路**

利用Object类自带的sort方法和传入Comparator类比较属性

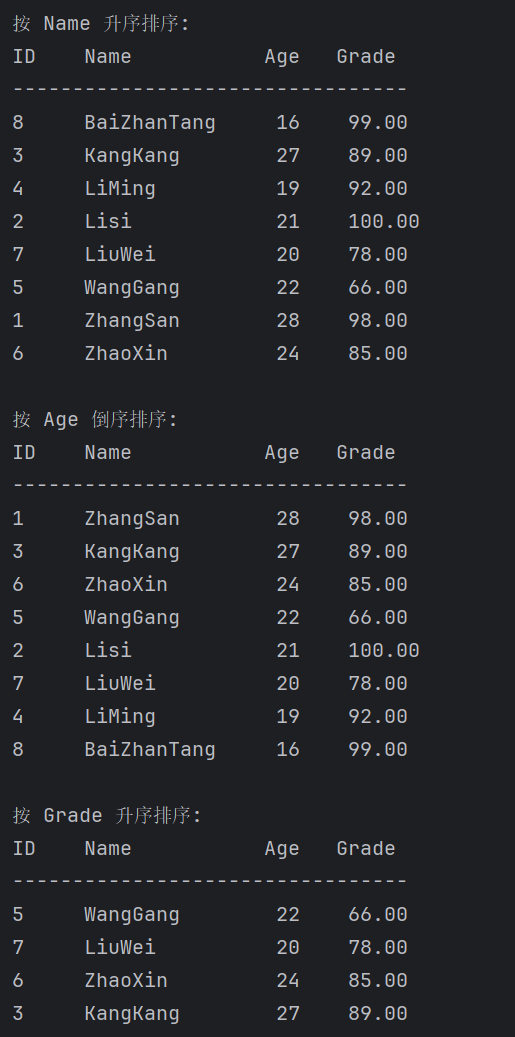
1. **实验源码**

**StudentSort**

class StudentSort {  
 private int id;  
 private String name;  
 private int age;  
 private double grade;  
  
 public StudentSort(int id, String name, int age, double grade) {  
 this.id = id;  
 this.name = name;  
 this.age = age;  
 this.grade = grade;  
 }  
  
 public int getId() { return id; }  
 public String getName() { return name; }  
 public int getAge() { return age; }  
 public double getGrade() { return grade; }  
  
 @Override  
 public String toString() {  
 return String.format("%-5d %-15s %-5d %-5.2f", id, name, age, grade);  
 }  
}

**TestStudentSort**

import java.util.\*;  
  
public class TestStudentSort {  
 public static void main(String[] args) {  
 List<StudentSort> students = new ArrayList<>();  
 students.add(new StudentSort(1, "ZhangSan", 28, 98));  
 students.add(new StudentSort(2, "Lisi", 21, 100));  
 students.add(new StudentSort(3, "KangKang", 27, 89));  
 students.add(new StudentSort(4, "LiMing", 19, 92));  
 students.add(new StudentSort(5, "WangGang", 22, 66));  
 students.add(new StudentSort(6, "ZhaoXin", 24, 85));  
 students.add(new StudentSort(7, "LiuWei", 20, 78));  
 students.add(new StudentSort(8, "BaiZhanTang", 16, 99));  
  
 // 按Name升序排序  
 students.sort(Comparator.comparing(StudentSort::getName));  
 printStudents("按 Name 升序排序:", students);  
  
 // 按Age倒序排序  
 students.sort(Comparator.comparing(StudentSort::getAge).reversed());  
 printStudents("按 Age 倒序排序:", students);  
  
 // 按Grade升序排序  
 students.sort(Comparator.comparing(StudentSort::getGrade));  
 printStudents("按 Grade 升序排序:", students);  
 }  
  
 private static void printStudents(String message, List<StudentSort> students) {  
 System.out.println("\n" + message);  
 System.out.println("ID Name Age Grade");  
 System.out.println("---------------------------------");  
 for (StudentSort s : students) {  
 System.out.println(s);  
 }  
 }  
}

****

1. **实验心得**

利用Object类自带的sort方法和传入Comparator类比较属性

实验三

1. **实验内容**

（1）编写构造方法和各数据成员的set、get方法。

（2）编写Student类的getHours()方法，计算学生选课的总学分。

（3）编写Student类中的showMessage()方法，打印输出学生信息（姓名、年级、所有选课的名称、学分以及总学分）。

（4）编写一个测试类，在控制台输入几门课程的信息，几个学生的信息，打印输出这些学生的选课信息。

1. **实验思路**

用一个map存储对应的课程和学分，利用Java的链式编程可以直接获得总学分

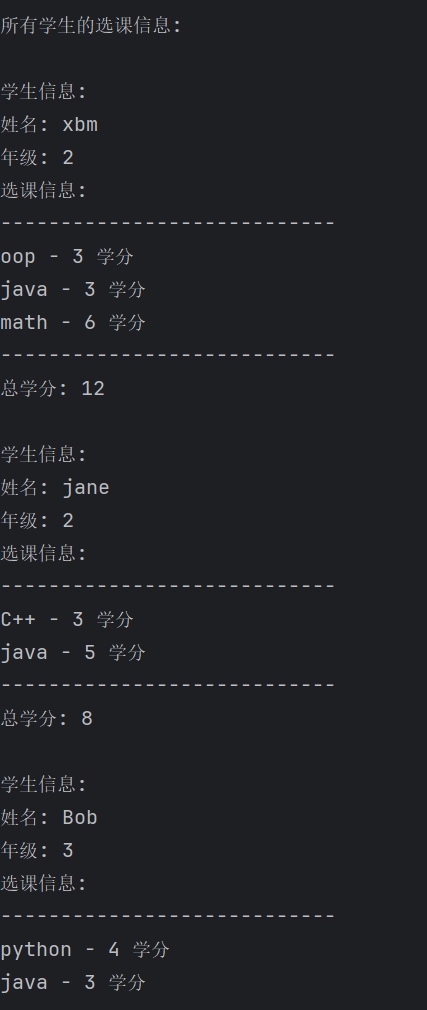
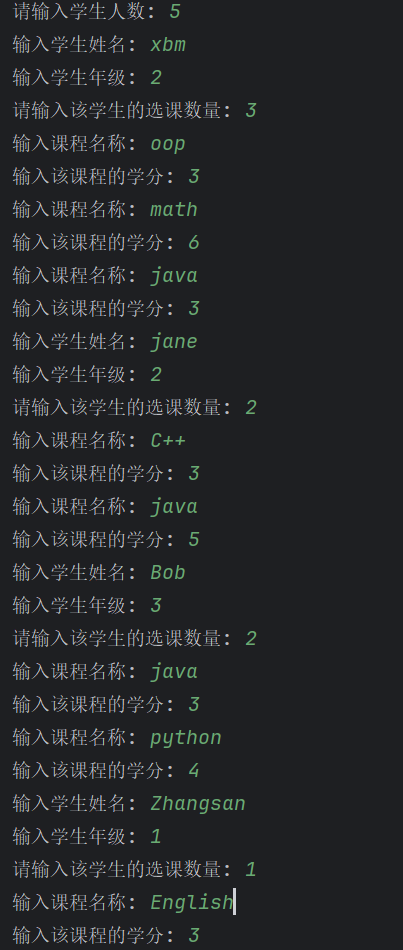
1. **实验源码**

包含Student和TestStudent两个类

**Student：**

import java.util.\*;  
class Students {  
 private String name;  
 private int grade;  
 private Map<String, Integer> courses; // 课程名称 -> 学分  
  
 public Students(String name, int grade) {  
 this.name = name;  
 this.grade = grade;  
 this.courses = new HashMap<>();  
 }  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 public int getGrade() { return grade; }  
 public void setGrade(int grade) { this.grade = grade; }  
  
 public void addCourse(String courseName, int credit) {  
 courses.put(courseName, credit);  
 }  
  
 public int getHours() {  
 return courses.values().stream().mapToInt(Integer::intValue).sum();  
 }  
  
 public void showMessage() {  
 System.out.println("\n学生信息:");  
 System.out.println("姓名: " + name);  
 System.out.println("年级: " + grade);  
 System.out.println("选课信息:");  
 System.out.println("----------------------------");  
 for (Map.Entry<String, Integer> entry : courses.entrySet()) {  
 System.out.println(entry.getKey() + " - " + entry.getValue() + " 学分");  
 }  
 System.out.println("----------------------------");  
 System.out.println("总学分: " + getHours());  
 }  
}

**TestStudent：**import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
  
public class TestStudent {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 List<Students> studentList = new ArrayList<>();  
  
 System.out.print("请输入学生人数: ");  
 int studentCount = scanner.nextInt();  
 scanner.nextLine(); // 清除换行符  
  
 for (int i = 0; i < studentCount; i++) {  
 System.out.print("输入学生姓名: ");  
 String name = scanner.nextLine();  
 System.out.print("输入学生年级: ");  
 int grade = scanner.nextInt();  
 scanner.nextLine();  
  
 Students student = new Students(name, grade);  
  
 System.out.print("请输入该学生的选课数量: ");  
 int courseCount = scanner.nextInt();  
 scanner.nextLine();  
  
 for (int j = 0; j < courseCount; j++) {  
 System.out.print("输入课程名称: ");  
 String courseName = scanner.nextLine();  
 System.out.print("输入该课程的学分: ");  
 int credit = scanner.nextInt();  
 scanner.nextLine();  
 student.addCourse(courseName, credit);  
 }  
  
 studentList.add(student);  
 }  
  
 System.out.println("\n所有学生的选课信息:");  
 for (Students student : studentList) {  
 student.showMessage();  
 }  
  
 scanner.close();  
 }  
}



1. **实验心得**

用好Java提供的工具