public class Person {
    protected String name;
    protected int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public void birthday() {
        age++;
    }
} // end class Person

public class Student extends Person implements Comparable<Student> {
    private String major;
    public Student(String name, int age, String major) {
        super(name, age);
        this.major = major;
    } // end constructor
    // Two students are considered equal if they have the same name
    public boolean equals(Object other) {
        if (other instanceof Student) {
            Student otherStudent = (Student) other;
            return (name == otherStudent.name);
        }
        else
            return false;
    } // end equals
    // The "natural order" compares students by name only
    public int compareTo(Student other) {
        return name.compareTo(other.name);
    } // end compareTo
} // end class Student

// A MajorComparator compares two students by their majors, then their ages
public class MajorComparator implements Comparator<Student> {
    public int compare(Student s1, Student s2) {
        int result = s1.major.compareTo(s2.major);
        if (result == 0) // use ages
            result = s1.age - s2.age;
        return result;
    } // end compare
} // end MajorComparator