

「グラフと組み合わせ」 課題 5 (解答例)

2010/5/17

1 Javaの利用

講義で示したプログラムでは、得点を使ってソートを行った。ソートに使用するキーを設定できるように変更することを考える。クラス `Student` に、ソートに使用するキーを定義し、メソッド `compareTo` 内の動作をそのキーに応じて変更することにする。

`Student` クラスの対応する部分は、以下のようになる。

```
public class Student implements java.lang.Comparable<Student> {
```

```
    public static enum SortKey {
        NAME, ID, RECORD;
    }
    private SortKey sortKey = SortKey.RECORD;
    /**
     * @return the sortKey
     */
    public SortKey getSortKey() {
        return sortKey;
    }

    /**
     * @param sortKey the sortKey to set
     */
    public void setSortKey(SortKey sortKey) {
        this.sortKey = sortKey;
    }

    @Override
    /**
     * Student インスタンスの比較
     * インターフェイス Comparable で必須

```

```

    */
    public int compareTo(Student o) {
        int k = 0;
        switch (sortKey) {
            case NAME:
                //名前の場合
                break;
            case ID:
                //学籍番号の場合
                break;
            default:
                //得点の場合
                break;
        }
        return k;
    }
}

```

対応するクラス **StudentRecord** では、ソートを実施する前に、ソートキーを設定することになる。

```

    public void setSortKey(Student.SortKey sortKey) {
        for (int i = 0; i < getStudents().length; i++) {
            Student s = getStudents()[i];
            s.setSortKey(sortKey);
        }
    }
}

```

メソッド **compareTo** 内の動作を実装し、ソートキーを変更し、ソートを実施しなさい。

解答例

別紙参照

Student.java

```
/*
 * Student.java
 *
 * Created on 2007/04/15, 11:34
 * 生徒のクラス
 */
package StudentSample4;

/**
 *
 * @author tadaki
 */
public class Student implements java.lang.Comparable<Student> {

    private String name = null; //名前
    private int studentID = 0; //学生番号
    private int record = 0; //点数

    public static enum SortKey {

        NAME, ID, RECORD;
    }

    private SortKey sortKey = SortKey.RECORD;

    /** コンストラクタ */
    public Student(String name, int studentID) {
        this.name = new String(name);
        this.studentID = studentID;
    }

    public boolean find(int s) {
        if (s == getStudentID()) {
            return true;
        }
        return false;
    }

    public int getStudentID() {
        return studentID;
    }

    public String getName() {
```

Student.java

```
        return name;
    }

    public int getRecord() {
        return record;
    }

    public void setRecord(int record) {
        this.record = record;
    }

    /**
     * @return the sortKey
     */
    public SortKey getSortKey() {
        return sortKey;
    }

    /**
     * @param sortKey the sortKey to set
     */
    public void setSortKey(SortKey sortKey) {
        this.sortKey = sortKey;
    }

    @Override
    /**
     * Student インスタンスの比較
     * インターフェイスComparableで必須
     */
    public int compareTo(Student o) {
        int k = 0;
        switch (sortKey) {
            case NAME:
                k=this.getName().compareTo(o.getName());
                break;
            case ID:
                if (this.getStudentID() > o.getStudentID()) {
                    k = 1;
                }
                if (this.getStudentID() < o.getStudentID()) {
                    k = -1;
                }
                break;
        }
    }
}
```

Student. java

```
        default:
            if (this.getRecord() > o.getRecord()) {
                k = 1;
            }
            if (this.getRecord() < o.getRecord()) {
                k = -1;
            }
            break;
    }
    return k;
}
```

StudentRecord.java

```
/*
 * StudentRecord.java
 *
 * Created on 2007/04/15, 11:34
 * パッケージを作って、関連するプログラムを一つにまとめる
 */
package StudentSample4;

/**
 *
 * @author tadaki
 */
public class StudentRecord {

    private Student students[] = null; //生徒一覧
    private String names[] = {
        "Aoyama", "Asou", "Baba", "Chou", "Egashira",
        "Eto", "Funaki", "Goto", "Gunji", "Hara", "Hashimoto",
        "Ikeuchi", "Ito", "Jo", "Kayama", "Mori", "Naito", "Tada",
        "Yamada", "Yoshida"
    };

    /** コンストラクタ */
    public StudentRecord() {
        //生徒一覧を初期化
        students = new Student[names.length];
        //登録
        for (int i = 0; i < names.length; i++) {
            Student s = new Student(names[i], 1000 + i);
            s.setRecord((int) (100 * Math.random()));
            students[i] = s;
        }
    }

    public void listStudents() {
        //拡張されたforループ
        for (int i = 0; i < getStudents().length; i++) {
            Student s = getStudents()[i];
            System.out.print(String.valueOf(s.getStudentID())
                + ":" + s.getName() + ":");
            System.out.println(String.valueOf(s.getRecord()));
        }
    }
}
```

StudentRecord.java

```
public Student[] getStudents() {
    return students;
}

public static <T extends java.lang.Comparable<T>> void sort(T t[]) {
    for (int i = t.length; i > 0; i--) {
        for (int j = 0; j < i - 1; j++) {
            if (t[j].compareTo(t[j + 1]) > 0) {
                T c = t[j];
                t[j] = t[j + 1];
                t[j + 1] = c;
            }
        }
    }
}

public void setSortKey(Student.SortKey sortKey) {
    for (int i = 0; i < getStudents().length; i++) {
        Student s = getStudents()[i];
        s.setSortKey(sortKey);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    StudentRecord studentRecord = new StudentRecord();
    studentRecord.setSortKey(Student.SortKey.NAME);
    StudentRecord.sort(studentRecord.getStudents());
    studentRecord.listStudents();
}
}
```