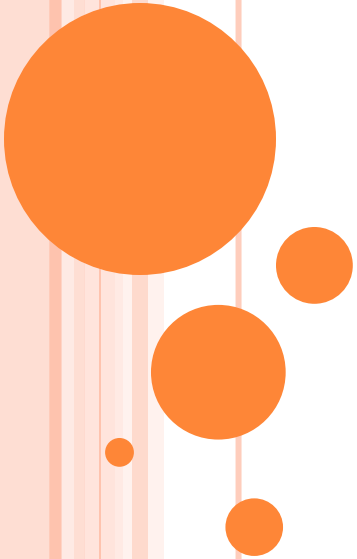


簡単なJavaプログラム その2

オブジェクト指向プログラミング特論

只木進一：総合情報基盤センター

前回のプログラムで発生しそうな不都合

- 通常、ソートは、数字を並べ替えるのが目的ではない。
  - データを何かの順に並べ替える
    - 順序が定められれば、何でもよい
- データの種類に対応して、コードを作り替えることになる
  - ソートは、標準的手法なのに

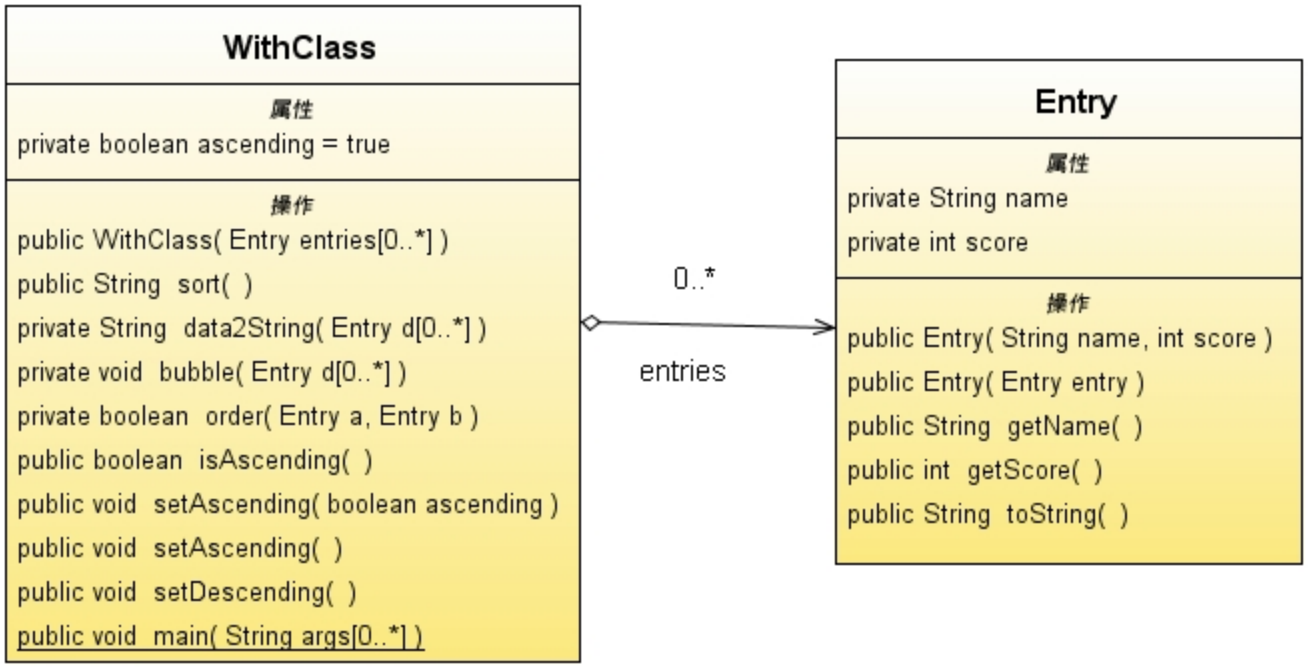
クラスを使って改善してみよう

- 名前と点数を保持するクラスEntry
  - Entry.java
- 新しいインスタンスの生成new
  - 配列を一度に作ることもできる

```
new Entry[]{
        new Entry("Bob", 90),
        new Entry("Mary", 70),
        new Entry("Tom", 95),
        new Entry("Mark", 85),
        new Entry("Betty", 80)
    }
```

- メインクラスの変更：WithClass.java
- 順序を確かめるメソッド

```
private boolean order(Entry a, Entry b) {
        boolean ans = false;
        if (this.isAscending() && (a.getScore() < b.getScore())) {
            ans = true;
        }
        return ans;
    }
```


WithClass
属性
private boolean ascending = true
操作
public WithClass(Entry entries[0..*])
public String sort( )
private String data2String(Entry d[0..*])
private void bubble(Entry d[0..*])
private boolean order(Entry a, Entry b)
public boolean isAscending( )
public void setAscending(boolean ascending)
public void setAscending( )
public void setDescending( )
public void main(String args[0..*])
0..*
entries
Entry
属性
private String name
private int score
操作
public Entry(String name, int score)
public Entry(Entry entry)
public String getName( )
public int getScore( )
public String toString( )

何が問題か・何を学ぶか

- メインのクラスがEntryクラスの中身を知らねばならない
  - 別のデータには別のプログラムが必要になる
- データの実装とデータを処理する過程を分離
  - クラスの抽象化
  - 抽象的データ構造
  - 抽象的インターフェイス
  - デザインパターン

JAVAにおける抽象クラス

- Abstract Class
  - クラスの原型・共通的構造
  - フィールドを持つ
  - 一部のメソッドが実装されていない
- Interface
  - 他のクラスからの呼ばれ方を定義
  - 定数と実装されていないメソッドだけを持つ

インターフェイスの利用

- java.lang.Comparable
  - 要素の比較を定義する抽象インターフェイス
  - 「比較できる」と実際の比較方法を分離
- 新しいEntryクラス：EntryNew.java
- 新しいWithClassNewクラス：WithClassNew.java

EntryNewクラス

- 大小関係を比較できるクラスとして宣言

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

EntryNewクラスのインスタンスと比較できる

- 比較のためのメソッド

```
/**
    * Comparableインターフェイスに必要な比較のメソッド
    * @param e 比較対象
    * @return 自分が大きければ1、小さければ-1、同じならば0
    */
   public int compareTo(EntryNew e) {
       if (e.getScore() > score) {
           return -1;
       }
       if (e.getScore() < score) {
           return 1;
       }
       return 0;
   }
```

WithClassNew

- java.lang.Comparableインターフェイスを持ったクラスらなば、なんでもソートできるクラス
  - 対象クラス名をテンプレートで表示
  - クラス名EntryNewは表れない！

```
public class WithClassNew<T extends Comparable<T>>
```