

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 11 次の Java プログラムの説明及びプログラムを読んで、設問 1～3 に答えよ。

(Java プログラムで使用する API の説明は、この冊子の末尾を参照してください。)

[プログラムの説明]

文字列の配列を要素にもつ配列で表現された表中の行を、並べ替えるプログラムである。表は行の配列として、行は文字列の配列として構成される。各行の指定した列が、指定した順序に並ぶように、行を並べ替える。

図 1 は、表の最左列の文字列を辞書順に、最左列の文字列が等しい行は右隣の列の文字列を数値とみなしたときに降順になるように並べ替えた例である。

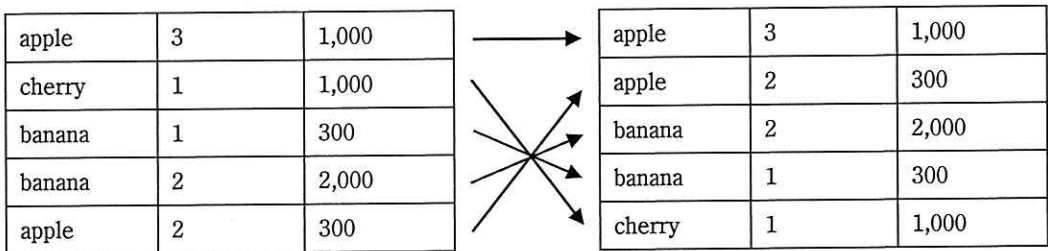


図 1 並べ替えの例

クラス `TableSorter` は、表中の行を並べ替える機能を提供する。次のメソッドと入れ子クラス `OrderBy` をもつ。

(1) `putSortOrder(String key, Comparator<String> order)` — このクラスのメソッド `sort` で指定可能な順序を登録する。

`key` — 表を並べ替える順序を示すために使用する文字列（以下、順序指定子という）。

`order` — インタフェース `Comparator` 型のオブジェクトで、`key` で指定した順序指定子が示す順序を規定する。

(2) `sort(String[][] table, final OrderBy... orderBys)` — 条件として指定した列が指定した順序になるように、表中の行を並べ替える。条件は複数指定できる。

並び順の決定には先に指定した条件が優先される。先に指定した条件で並び順が決まらない行については、後に指定した条件を加えて決定する。指定した全ての条件で順序が決まらない行については、並べ替える前の並び順を維持する。

`table` — 整列対象の表を格納した `String` 型の配列を要素にもつ配列を指定する。
`orderBy` — 条件を必要なだけ指定する。

クラス `TableSorter` の入れ子クラス `OrderBy` は並べ替えの条件を表すクラスである。フィールド `key` は順序指定子、フィールド `col` は比較対象の列の位置を示す数（行を表す `String` 型配列の添字）である。フィールド `isReversed` が `false` であれば、この条件が表す順序が順序指定子 `key` で指定されるオブジェクトのメソッド `compare` が返す大小関係での昇順で並べ替えることを意味し、`true` であれば、降順で並べ替えることを意味する。

クラス `TableSorterTester` はテスト用のプログラムである。
実行結果を図 2 に示す。

```
apple 3 1,000
apple 2 300
banana 2 2,000
banana 1 300
cherry 1 1,000
```

図 2 実行結果

[プログラム 1]

```
import java.util.Arrays;
import java.util.Comparator;
import java.util.HashMap;
import java.util.Map;

public class TableSorter {
    private Map<  > orderMap = new HashMap<  >();

    public void putSortOrder(String key, Comparator<String> order) {
        orderMap.put(key, order);
    }

    public void sort(String[][] table, final OrderBy... orderBys) {
        Arrays.sort(table, new Comparator<String[]>() {
            public int compare(String[] s1, String[] s2) {
                for (  ) {
                    int order = orderMap.get(orderBy.key).
                        compare(s1 , s2 );
                    if (order  0) {
                        return ;
                    }
                }
                return 0;
            }
        });
    }

    public static class OrderBy {
        final String key;
        final int col;
        final boolean isReversed;

        public OrderBy(String key, int col) {
            this(key, col, false);
        }

        public OrderBy(String key, int col, boolean isReversed) {
            this.key = key;
            this.col = col;
            this.isReversed = isReversed;
        }
    }
}
```

```
    }  
  }  
}
```

[プログラム 2]

```
import java.util.Comparator;  
  
public class TableSorterTester {  
    public static void main(String... args) {  
        String[][] data = new String[][] {  
            {"apple", "3", "1,000"},  
            {"cherry", "1", "1,000"},  
            {"banana", "1", "300"},  
            {"banana", "2", "2,000"},  
            {"apple", "2", "300"},  
        };  
        TableSorter sorter = new TableSorter();  
        sorter.putSortOrder("lex", new Comparator<String>() {  
            public int compare(String o1, String o2) {  
                return o1.compareTo(o2);  
            }  
        });  
        sorter.putSortOrder("num", new Comparator<String>() {  
            public int compare(String o1, String o2) {  
                return new Integer(o1).compareTo(new Integer(o2));  
            }  
        });  
        sorter.sort(data, new TableSorter.OrderBy("lex", 0),  
                    new TableSorter.OrderBy("num", 1, true)); } ← α  
        for (String[] row : data) {  
            for (String col : row) {  
                System.out.printf("%s ", col);  
            }  
            System.out.println();  
        }  
    }  
}
```

設問1 プログラム1中の に入れる正しい答えを、解答群の中から選べ。

aに関する解答群

- ア Character, Comparator<String>
- イ Character, String
- ウ String, Comparator<String>
- エ String, String

bに関する解答群

- ア `int i = 0; i < orderBys.length; i++`
- イ `int i = 0; i < table.length; i++`
- ウ `OrderBy orderBy : orderBys`
- エ `String[] row : table`

cに関する解答群

- ア `.get(col)`
- イ `.get(orderBy.col)`
- ウ `[col]`
- エ `[orderBy.col]`

dに関する解答群

- ア `!=`
- イ `<`
- ウ `==`
- エ `>`

eに関する解答群

- ア `-order`
- イ `order`
- ウ `orderBy.isReversed ? -order : order`
- エ `orderBy.isReversed ? order : -order`

設問2 プログラム2のαで示した2行を次の2行と入れ替えて実行したとき、実行結果の1行目に出力される内容として正しい答えを、解答群の中から選べ。

```
sorter.sort(data, new TableSorter.OrderBy("lex", 2),
            new TableSorter.OrderBy("lex", 0));
```

解答群

- | | | |
|------------------|------------------|----------------|
| ア apple 2 300 | イ apple 3 1,000 | ウ banana 1 300 |
| エ banana 2 2,000 | オ cherry 1 1,000 | |

設問3 プログラム2のαで示した2行を次の8行と入れ替えて実行したとき、実行結果の3行目に出力される内容として正しい答えを、解答群の中から選べ。

```
sorter.putSortOrder("numC", new Comparator<String>() {
    public int compare(String s1, String s2) {
        return new Integer(s1.replace(",","")).
            compareTo(new Integer(s2.replace(",","")));
    }
});
sorter.sort(data, new TableSorter.OrderBy("numC", 2),
            new TableSorter.OrderBy("lex", 0));
```

解答群

- | | | |
|------------------|------------------|----------------|
| ア apple 2 300 | イ apple 3 1,000 | ウ banana 1 300 |
| エ banana 2 2,000 | オ cherry 1 1,000 | |