Java 基礎問題-02-02-08

Java 基礎問題-02-02-06 の従業員表(Emp)クラスの
全メソッド・フィールドで static を使用するように修正してください。従業員表(Emp)クラスのインスタンスはEmpクラス外からnewできないように、従業員表(Emp)クラスのコンストラクタはprivete にしてください。

従業員表(Emp)クラスに selectAllOrderBySal(boolean desc) を新規追加してください。

KadaiStaticEmp8.java に main()メソッドを作成し動作を確認してください。

レコード(Row)
int empno String ename String job int sal
getter/setter

従業員表(Emp)
static Row[] rowArray static int insIndex
static void insert(Row row) static int updateByEmpno(int empno, Row updateRow) static int deleteByEmpno(int empno) static Row[] selectAll() static int[] selectAllEmpno() static Row selectByPrimaryKey(int empno) static Row[] selectByName(String ename) static int selectMaxSal() static int selectRowCount() static Row[] selectAllOrderBySal(boolean desc)

ヒント

▼必要とされる知識
オブジェクトの配列の概念
オブジェクトの参照渡し
オブジェクトの参照を返すメソッド

▼考え方のポイント
Java クラスモデリング問題 6 を使用し、効率的に実装を進めましょう。
新規追加メソッドの実装内容などは no008 パッケージに格納されている Java ファイルの Javadoc コメント(/** */)に記載されていますので、そちらを参照し実装していきましょう。

・selectAllOrderBySal(boolean desc)は、例えば
アルゴリズム「バブルソート」を使用すると実現できます。

実行結果 ※数値と記号は半角を利用してください。

※以下実行結果の通りコンソールに出力されるように実装しましょう。
クラスモデリング問題 06 と差異がある部分を赤字にしています。

```
═══ 従業員表にレコードを追加する(insert(Row row)メソッド) ═══
すみすさんの情報が一行挿入されました
ありすさんの情報が一行挿入されました
主キーが重なっているので挿入できません
従業員番号に0は挿入できません
けんさんの情報が一行挿入されました
めありーさんの情報が一行挿入されました
けんさんの情報が一行挿入されました
もう挿入できません
═══ 従業員表にレコードを追加しました(insert(Row row)メソッド) ═══


═══ 今の従業員表登録件数を表示開始(selectRowCount()メソッド) ═══
今従業員表には5件登録されています。
═══ 今の従業員表登録件数を表示終了(selectRowCount()メソッド) ═══


═══ 指定した従業員番号に該当するレコードを削除開始(deleteByEmpno(int empno)メソッド) ═══
削除された行は0
削除された行は1
═══ 指定した従業員番号に該当するレコードを削除終了(deleteByEmpno(int empno)メソッド) ═══
```

```
========== 今の従業員表登録件数を表示開始(selectRowCount()メソッド) ==========
今従業員表には4件登録されています。
========== 今の従業員表登録件数を表示終了(selectRowCount()メソッド) ==========

========== 指定した従業員番号に該当するレコードを更新開始(updateByEmpno(int empno, Row updateRow)メソッド) ==========
更新された行は0
更新行数は1
========== 指定した従業員番号に該当するレコードを更新終了(updateByEmpno(int empno, Row updateRow)メソッド) ==========

========== 従業員名で検索開始(selectByName(String ename)メソッド) ==========
従業員番号:3333,名前 : けん,役職:一般社員
従業員番号:5555,名前 : けん,役職:派遣社員
========== 従業員名で検索終了(selectByName(String ename)メソッド) ==========

========== 従業員表に登録されている従業員中、最も高い給料検索開始(selectMaxSal()メソッド) ==========
一番高いお給料は8400
========== 従業員表に登録されている従業員中、最も高い給料検索終了(selectMaxSal()メソッド) ==========

========== 指定した従業員番号に当てはまるレコード検索開始(selectByPrimaryKey(int empno)メソッド) ==========
従業員番号:3333,名前 : けん,役職:一般社員
========== 指定した従業員番号に当てはまるレコード検索終了(selectByPrimaryKey(int empno)メソッド) ==========

========== 従業員表に登録されている全レコード取得開始(selectAll()メソッド) ==========
従業員番号:1111,名前 : すみす,役職:一般社員
従業員番号:2222,名前 : NEWありす,役職:受付係
従業員番号:3333,名前 : けん,役職:一般社員
従業員番号:5555,名前 : けん,役職:派遣社員
========== 従業員表に登録されている全レコード取得終了(selectAll()メソッド) ==========

========== 従業員表に登録されている全レコードの従業員番号一覧取得開始(selectAllEmpno()メソッド) ==========
1111
2222
3333
5555
========== 従業員表に登録されている全レコードの従業員番号一覧取得終了(selectAllEmpno()メソッド) ==========
```

```
======= 給料で降順ソートした全レコードを取得開始(selectAllOrderBySal(true)メソッド) =======
従業員番号:3333, 名前 : けん, 役職:一般社員, 給料8400
従業員番号:1111, 名前 : すみす, 役職:一般社員, 給料800
従業員番号:5555, 名前 : けん, 役職:派遣社員, 給料800
従業員番号:2222, 名前 : NEWあります, 役職:受付係, 給料500
======= 給料で降順ソートした全レコードを取得終了(selectAllOrderBySal(true)メソッド) =======

======= 給料で昇順ソートした全レコードを取得開始(selectAllOrderBySal(false)メソッド) =======
従業員番号:2222, 名前 : NEWあります, 役職:受付係, 給料500
従業員番号:1111, 名前 : すみす, 役職:一般社員, 給料800
従業員番号:5555, 名前 : けん, 役職:派遣社員, 給料800
従業員番号:3333, 名前 : けん, 役職:一般社員, 給料8400
======= 給料で昇順ソートした全レコードを取得終了(selectAllOrderBySal(false)メソッド) =======
```