

# Ex\_08\_02.java 解説

このプログラムは、学生データをファイルから読み込み、各学生の合計点数に基づいて順位を決定し、上位3位以内にランクインするすべての学生（同点の場合を含む）を出力するJavaプログラムです。以下に、各部分の詳細な解説を行います。

## 1. ファイルの読み込み

```
String fileName = args[0];
List<String[]> students = new ArrayList<>();

// ファイルを読み込み
try (BufferedReader br = new BufferedReader(new FileReader(
    fileName))) {
    String line;
    while ((line = br.readLine()) != null) {
        String[] data = line.trim().split("\\\\s+");
        students.add(data);
    }
} catch (IOException e) {
    System.out.println("ファイルを読み込めません: " + e.getMessage());
    return;
}
```

- `String fileName = args[0];`: コマンドライン引数から、読み込むファイルの名前を取得します。
- `List<String[]> students = new ArrayList<>();`: 学生データを格納するリストを作成します。各学生のデータは文字列の配列 (`String[]`) として格納されます。
- **ファイルの読み込み**: `BufferedReader` を使用して、指定されたファイルを1行ずつ読み込みます。各行のデータは空白で区切られており、`split("\\\\s+)` を使用して分割されます。分割されたデータは `students` リストに追加されます。

## 2. 学生データのソート

```

Collections.sort(students, (s1, s2) -> {
    int score1 = Integer.parseInt(s1[2]) + Integer.parseInt(
s1[3]) + Integer.parseInt(s1[4]);
    int score2 = Integer.parseInt(s2[2]) + Integer.parseInt(
s2[3]) + Integer.parseInt(s2[4]);
    return Integer.compare(score2, score1) != 0 ? Integer.c
ompare(score2, score1) : s1[0].compareTo(s2[0]);
});

```

- `Collections.sort()` : `students` リストをソートします。ソートの基準は、まず合計点数（降順）で、それが同じ場合は学生ID（昇順）でソートします。
- `int score1, score2` : 各学生の合計点数を計算します。 `s1[2], s1[3], s1[4]` はそれぞれ3科目の点数を表しています。
- `Integer.compare()` : 2つの整数値を比較します。合計点数が異なる場合は、その差に基づいてソートし、同じ場合は学生IDの辞書順にソートします。

### 3. 上位3位以内の学生を出力

```

int rank = 1;
int count = 0;
int previousScore = -1;
for (String[] student : students) {
    int totalScore = Integer.parseInt(student[2]) + Integer.parseInt(student[3]) + Integer.parseInt(student[4]);

    if (totalScore != previousScore) {
        rank = count + 1; // 新しい順位を設定
    }

    if (rank > 3) {
        break; // 3位以内の学生のみを表示
    }

    System.out.printf("%s,%s,%d\\n", student[0], student[1], totalScore);
    previousScore = totalScore;
}

```

```
        count++;  
    }
```

- `int rank = 1;` : 現在の順位を保持するための変数です。最初は1位から開始します。
- `int count = 0;` : 処理済みの学生数をカウントするための変数です。
- `int previousScore = -1;` : 前の学生の合計点数を保持するための変数です。初期値は-1です。
- **for ループ:** `students` リストを順に処理し、各学生の合計点数を計算します。
- `if (totalScore != previousScore)` : 現在の学生の合計点数が前の学生と異なる場合、新しい順位を設定します (`rank = count + 1`) 。
- `if (rank > 3)` : 順位が3位を超えた場合、ループを終了し、それ以降の学生は出力されません。
- `System.out.printf("%s,%s,%d\\n", ...)` : 現在の学生のID、名前、合計点数をフォーマットして出力します。