

課題 8

複数条件式の利用

順番にキーボードから入力された100m走のタイム(double型の小数)を受け取り, 新しいタイムが入力された際にタイムを比較しながら順位を入れ替えて, タイムが早い順に第3位までを表示できるクラス**Top3Records** を作成しましょう.

課題 8

仕様

- フィールド topRecord, secondRecord, thirdRecord
 - 3つのタイムを早い順に格納する(double型)
- コンストラクタ
 - double型の引数を1つ渡すと, 3つのフィールドを引数で与えられたタイムで初期化する
- registerNewRecord()
 - 新しいタイム1つ(double型)を引数とし、それらと topRecord, secondRecord, thirdRecord を比較して順位を入れ替えるメソッド(void型)
- getTopRecord () topRecord の値を返すメソッド(double型)
- showAllRecords() 一位から三位までのタイムを表示するメソッド(void型)

クラス図

Top3Records
-topRecord: double -secondRecord: double -thirdRecord: double
+Top3Records(initialRecord: double) +registerNewRecord(newRecord: double): void +getTopRecord(): double +showAllRecords(): void <u>+main(args: String[]): void</u>

課題8の実行例

100m走のタイムを1つ入力してください: 10.0

一位=10.00, 二位=10.00, 三位=10.00

100m走のタイムをスペース区切りで3つ入力してください:

9.72 9.99 11.0

一位=9.72, 二位=9.99, 三位=10.00

最高タイムを初期値として新しい順位表を作成します.

一位=9.72, 二位=9.72, 三位=9.72

タイムを1つ入力してください: 9.58

さらにタイムを1つ入力してください: 9.84

最終結果:

一位=9.58, 二位=9.72, 三位=9.72

水色の部分はキーボード入力

課題8の実行例2

100m走のタイムを1つ入力してください: 10.2

一位=10.20, 二位=10.20, 三位=10.20

100m走のタイムをスペース区切りで3つ入力してください:

10.9 10.6 9.9

一位=9.90, 二位=10.20, 三位=10.20

最高タイムを初期値として新しい順位表を作成します.

一位=9.90, 二位=9.90, 三位=9.90

タイムを1つ入力してください: 9.5

さらにタイムを1つ入力してください: 9.8

最終結果:

一位=9.50, 二位=9.80, 三位=9.90

水色の部分はキーボード入力

課題8の実行例3

100m走のタイムを1つ入力してください: 10.0

一位=10.00, 二位=10.00, 三位=10.00

100m走のタイムをスペース区切りで3つ入力してください:

11.0 12.0 13.0

一位=10.00, 二位=10.00, 三位=10.00

最高タイムを初期値として新しい順位表を作成します.

一位=10.00, 二位=10.00, 三位=10.00

タイムを1つ入力してください: 9.0

さらにタイムを1つ入力してください: 12.0

最終結果:

一位=9.00, 二位=10.00, 三位=10.00

水色の部分はキーボード入力

プログラムの骨格1

```
//スキャナクラスのインポート
```

```
クラス宣言 {
```

```
    // フィールドの宣言
```

```
    // コンストラクタの定義 {
```

```
        //仕様通りに記述する
```

```
    }
```

```
    // registerNewRecordメソッドの定義 {
```

```
        /** 新しいタイムを引数に与えることで、現在のタイムと比較してフィールドを入れ替えるメソッド
```

```
        * 例として、topRecordよりも小さな値が引数で与えられた場合、topRecordが引数で与えられた
```

```
        * 変数に置き換わる.
```

```
        * その際、元のtopRecordがsecondRecordに、元のsecondRecordがthirdRecordに置き換わる. */
```

```
    }
```

```
    // getTopRecordメソッドの定義 {
```

```
    // トップレコードを double型で返すメソッド
```

```
    }
```

```
/** 現在のタイムを順位通り表示するメソッド. 以下のまま利用すれば実行結果スライドにある出力ができる*/
```

```
public void showAllRecords(){
```

```
    System.out.printf("一位=%.2f, 二位=%.2f, 三位=%.2f ¥n",topRecord, secondRecord, thirdRecord);
```

```
}
```

```
public static void main(String[] args) { // プログラムの骨格2を参照してください
```

```
}
```

```
}
```

プログラムの骨格2 (Mainメソッドの中身)

```
public static void main(String[] args) {  
    // キーボード入力のためのスキャナオブジェクト生成  
  
    System.out.print("100m走のタイムを1つ入力してください: ");  
    // キーボードからの入力をdouble型の変数inputRecordに代入する  
  
    // inputRecordを初期値としてTop3Recordsクラスのオブジェクトを生成し、変数 recordBoard1で参照する  
    // recordBoard1 に各フィールドの値を表示させる(showAllRecordsメソッドを用いる)  
  
    System.out.print("タイムをスペース区切りで3つ入力してください: ");  
    // キーボードからの入力を double型の変数 record1, record2, record3 に代入する  
    // recordBoard1にregisterNewRecord を3回実行させ、record1, record2, record3 を登録する  
    // recordBoard1 に各フィールドの値を表示させる(showAllRecordsメソッドを用いる)  
  
    System.out.print("最高タイムを初期値として新しい順位表を作成します. ");  
    // double型の変数 topRecordを準備し、recordBoard1の最高タイムを代入する  
    // topRecordを初期値として新たに Top3Recordsクラスのオブジェクト recordBoard2 を生成する  
    // recordBoard2 に各フィールドの値を表示させる(showAllRecordsメソッドを用いる)  
  
    System.out.print("タイムを1つ入力してください: ");  
    // double型の変数 newRecord を準備し、新しい数値をキーボード入力から受け取る  
    // recordBoard2に registerNewRecord を実行させ、新しい数値を登録する  
    System.out.print("さらにタイムを1つ入力してください: ");  
    // newRecord が新しい数値をキーボード入力から受け取る  
    // recordBoard2 に registerNewRecord を実行させ、新しい数値を登録する  
  
    System.out.println("最終結果: ");  
    // recordBoard2 に各フィールドの値を表示させる(showAllRecordsメソッドを用いる)  
}
```