

Algorithmen und Datenstrukturen

Generics und Streams

Aufgabe 1 Competitor – 2 Punkte

Die Klasse `Competitor` haben Sie als Gerüst vorgegeben. Sie muss noch ergänzt werden, so dass die Teilnehmer nach der Zeit verglichen werden können.

Hinweise:

- Die `Competitor` Klasse muss das `Comparable<Competitor>` Interface implementieren
- Die `compareTo` Methode muss die Zeit als Ordnungskriterium nehmen wobei `parseTime` Methode verwendet werden kann.
- Sie sollten auch noch die `equals` und ev. `hashCode` Methode passend überschreiben.

Abgabe

Praktikum: ADS3.1

Filename: `Competitor.java`

Aufgabe 2 RankingListServer – 2 Punkte

Erstellen Sie einen `RankingListServer`, der die Rangliste einliest und eine generische Liste vom (Interface-)Typ `List<Competitor>` mit passender Implementation erstellt. Dazu muss die `createList` Methode implementiert werden.

Hinweise

- Verwenden Sie das Gerüst
- Nur die `RankingListServer` Klasse muss abgegeben werden; es kann von einer existierender, korrekten Implementation von `Competitor` ausgegangen werden.
- Mittels des `File→Open` Menüeintrags der `ExBox` können Sie den Inhalt einer Datei der `execute` Methode als `String` übergeben.

Abgabe

Praktikum: ADS3.2

Filename: `RankingListServer.java`

Aufgabe 3 Rangliste als Listen Implementation – 3 Punkt

Erstellen Sie eine Rangliste, d.h. sortieren Sie die Einträge nach der Zeit und geben Sie diese als `String` aus. Der Rang des Teilnehmers soll gesetzt sein. Dies soll in der `createSortedText` Methode implementiert werden. Es soll eine Textausgabe (`String`) mit der Rangliste wie unten dargestellt erstellt werden.

Hinweise

- Nur die RankingListServer Klasse muss abgegeben werden; es kann von einer existierender, korrekten Implementation von Competitor ausgegangen werden.
- Ausgabe:

```
Rangliste (List)
1 Kiptum Daniel 02:11:31
2 Ancay Tarcis 02:20:02
3 Kreibuhl Christian 02:21:47
4 Menzi Christoph 02:27:26
5 Marti Adrian 02:30:09
6 Mueller Stefan 02:31:14
7 Oliver Ruben 02:32:12
8 Kuehni Martin 02:33:36
9 Ott Michael 02:33:48
10 Elmer Beat 02:33:53
```

Abgabe

Praktikum: ADS3.3

Filename: RankingListServer.java

Aufgabe 4 Rangliste als Stream Implementation – 3 Punkte

Lösen Sie dieselbe Aufgabe mit den Stream Klassen und Funktionen (Lambda Ausdrücken). Dafür wird der Text zuerst in der `createStream` Methode in einen Strom von Teilnehmern umgewandelt, der dann in der `createSortedText` Methode sortiert und wieder zu einem Text umgewandelt wird.

Hinweise

- Nur die RankingStreamServer Klasse muss abgegeben werden; es kann von einer existierender, korrekten Implementation von Competitor ausgegangen werden.
- Ausgabe:

```
Rangliste (Stream)
1 Kiptum Daniel 02:11:31
2 Ancay Tarcis 02:20:02
3 Kreibuhl Christian 02:21:47
4 Menzi Christoph 02:27:26
5 Marti Adrian 02:30:09
6 Mueller Stefan 02:31:14
7 Oliver Ruben 02:32:12
8 Kuehni Martin 02:33:36
9 Ott Michael 02:33:48
10 Elmer Beat 02:33:53
```

Abgabe

Praktikum: ADS3.4

Filename: RankingStreamServer.java