

Algorithmen und Datenstrukturen

Generizität, Comparable und Comparator

Aufgabe 1: Competitor (Comparable-Interface)

Die Klasse Competitor haben Sie als Gerüst vorgegeben. Sie muss noch ergänzt werden, so dass die Teilnehmer nach der Zeit verglichen werden können. Führen Sie anschliessend den Test ADS3_1_test aus (sie brauchen die ExBox noch nicht).

Hinweise:

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

Abgabe:

Praktikum: ADS3.1

Filename: Competitor.java

Aufgabe 2: RankingListServer (generische Liste)

Erstellen Sie einen RankingListServer, der die Rangliste einliest (beigelegtes Textfile) und eine generische Liste vom (Interface-)Typ List<Competitor> mit passender Implementation (LinkedList oder ArrayList?) erstellt. Dazu muss die createList-Methode im Gerüst des RankingListServer implementiert werden. Zur Kontrolle geben Sie die eingelesene, unsortierte Liste wieder aus. Hinweis: StringBuffer (oder StringBuilder) und einen Iterator (oder foreach) verwenden.

Hinweis:

- 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

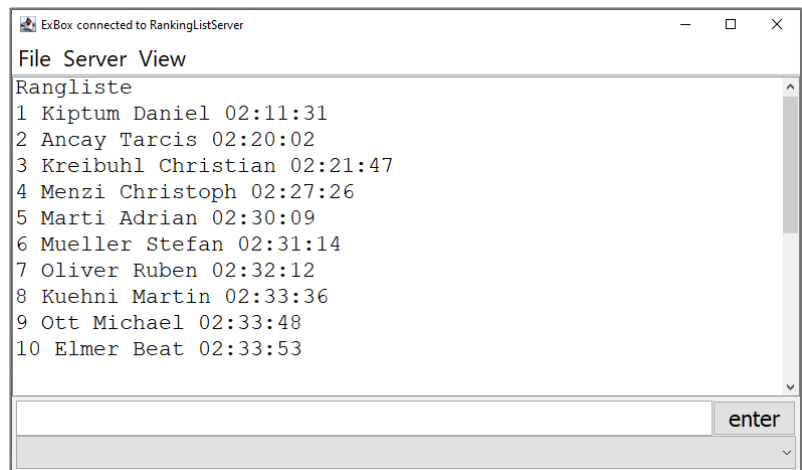
Aufgabe 3: Rangliste (Collections)

Erstellen Sie eine Rangliste, d.h. sortieren Sie die Einträge nach der Zeit und geben Sie diese als String aus (Zeilenumbrung mit «\n»). 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:

Abgabe:

Praktikum: ADS3.3

Filename: RankingListServer.java



```
ExBox connected to RankingListServer
File Server View
Rangliste
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
enter
```

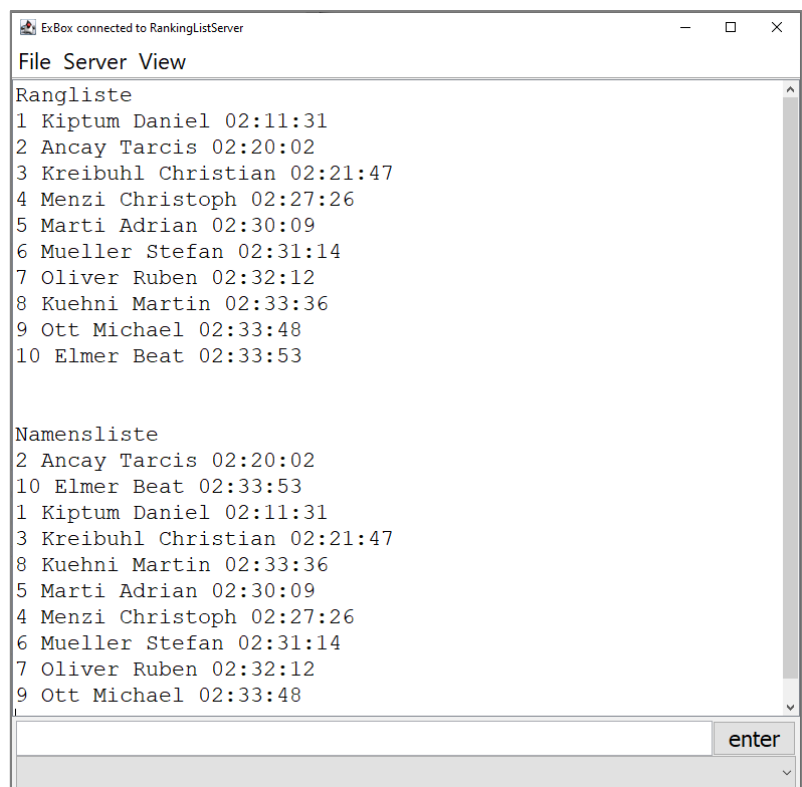
Aufgabe 4: Namensliste (Comparator)

Erstellen Sie eine Ausgabe, in der die Teilnehmer nach dem Namen sortiert sind. Bei gleichem Namen sollen sie nach Zeit sortiert werden (tieferer Zeit zuerst) Hinweis: Sie müssen dafür einen Comparator<Competitor> implementieren und die Collections.sort-Methode aufrufen. Dies soll in der createNameList-Methode implementiert werden. Es soll eine Textausgabe (String) mit der Rangliste wie unten dargestellt erstellt werden.

Abgabe:

Praktikum: ADS3.4

Filename: RankingListServer.java



```
ExBox connected to RankingListServer
File Server View
Rangliste
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

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