

Algorithmen und Datenstrukturen

Generics

Aufgabe 1 Competitor – 2 Punkte

Die Klasse `Competitor` haben Sie als Gerüst vorgegeben. Die Zeit wird intern in Millisekunden als `long` verwaltet. Sie können die Hilfsmethode `parseTime` für die Umwandlung aus einem String verwenden.

Aufgabe 2 RankingServer – 3 Punkte

Erstellen Sie einen `RankingServer`, der die Liste der Schweizer Teilnehmer des Zürich Marathons einliest und eine generische Liste vom Typ `List<Competitor>` mit passender Implementation erstellt (als Instanzvariablen des `RankingServers`). Hinweis: Mittels des `File` → `Open` Menüeintrags der `ExBox` können Sie den Inhalt einer Datei der `execute` Methode übergeben. Zur Kontrolle geben Sie die eingelesene Liste wieder aus. Hinweis: `StringBuffer` (oder `StringBuilder`) und einen Interator (oder `foreach`) verwenden.

http://openbook.galileocomputing.de/javainsel9/javainsel_04_004.htm#mj83f6ed07e8fdfe6af14e18edf6218235

Aufgabe 3 Rangliste – 2 Punkt

Erstellen Sie eine Rangliste, d.h. sortieren Sie die Einträge nach der Zeit und geben Sie diese aus. Hinweis: `Collections.sort` Methode verwenden. Die `Competitor` Klasse muss dafür das `Comparable<Competitor>` Interface implementieren und die `compareTo` Methode muss die Zeit als Ordnungskriterium nehmen. Sie sollten auch noch die `equals` Methode überschreiben.

Nr	Name, Vorname	JG	Wohnort	Zeit
4	Kiptum Daniel	1974	Magglingen	02:11:05
51	Ançay Tarcis	1974	Würenlos	02:10:58
...				

Aufgabe 4 Namensliste – 3 Punkte

Erstellen Sie eine Ausgabe, in der die Teilnehmer nach dem Namen sortiert sind. Bei gleichem Namen sollen sie nach Jahrgang sortiert werden (tiefe Jahrgänge zuerst) Hinweis: Sie müssen dafür einen `Comparator<Competitor>` implementieren und die `Collections.sort` aufrufen.