

Übung Stadtlauf

Beschreibung

Kompetenzen:

Die Stadtlauf Übung ist aus einer Teil-Prüfung der Informatiker aus dem Jahre 1999 entstanden. Inhalt der Übung ist die Handhabung von Dateien, Daten und Strukturen. Die Daten sollen aus Dateien eingelesen werden und in geeigneten Strukturen



abgelegt werden. Hierfür sollen entsprechende Klassen erstellt werden deren Objekte die Daten aufnehmen können. Die Daten sollen dann sortiert und in neue Dateien gespeichert werden.

Zeitbedarf: 5 Lektionen Hilfsmittel: Eclipse mit Java

Methode/Sozialform: EA 🚣

1. Text Files zeilenweise lesen und schreiben

2. Datenobjekte erstellen und Daten speichern

3. Strukturierte Daten sortieren

Abgaben:

Einzelarbeit, Partnerarbeit, Dokumentation, Code





Inhaltsverzeichnis

1	1 AUSGANGSLAGE		2
2	2 ZIEL DER AUFGABE	N	3
	2.1 LISTE MIT RESULTATE	EN	3
	2.1.1 Aufbau der Ra	anglisten-Dateien (junioren.rl, senioren.rl, elite.rl)	3
		amen-Liste (namen.ref)	
3	3 AUFGABEN		4
	3.1 DATENANALYSE		4
		D DEFINIEREN	
	3.3 ALGORITHMUS		4
4	4 ABGABEN		4
	4.2 DIACDAMME LIND NO	TIZEN	4

Quelle: www.senn-teaching.ch



1 Ausgangslage

Am Stadtlauf nehmen Läuferinnen und Läufer in verschiedenen Kategorien teil. Die LäuferInnen starten zu vordefinierten Startzeiten und kommen zu unterschiedlichen Zeiten im Ziel an. Ein paar Läuferinnen und Läufer kommen leider nicht im Ziel an.

Die Läuferinnen und Läufer erhalten je eine Startnummer und eine Kategorie in welcher sie Starten.

Am Stadtlauf starten die rund 170 gemeldeten Läufer in drei Kategorien:

```
1: Junioren (Start 14:00)
2: Senioren (Start 14:15)
3: Elite (Start 15:00)
```

Es existiert eine unsortierte Textdatei **startliste.txt** mit folgendem Aufbau:

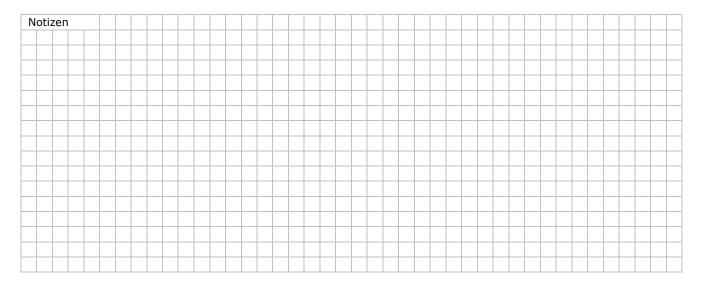
```
Start-Nr <tab> Kategorie <tab> Name
131 1 Gähwiler
072 3 Müller
256 1 Zehnder
012 2 Bucher
```

Die Namen sind max. 20 Zeichen lang, die Startnummer 1- bis 3-stellig von 1 bis 270 mit Lücken. Die nicht gebrauchten Nummern (Lücken) werden durch das Programm ignoriert. Das Zielgerät liefert die Einlaufzeiten (Tageszeit, chronologisch) aller eingetroffenen Läufer in der Datei **messresultate.txt**

```
Start-Nr <blank> Einlaufzeit 072 15:37:54 131 15:38:17
```

Bemerkung: die Daten sind "sauber": Keine zu langen Namen, keine Messresultate von nicht gemeldeten Läufern, keine negativen Laufzeiten, nur Ziffern für Zahlen. Das Programm muss also die Daten nicht auf ihre Gültigkeit überprüfen.

Achtung: Nicht alle gemeldeten Läufer sind am Ziel angekommen! In den Ranglisten werden nur die am Ziel angekommenen Läufer erfasst.





2 Ziel der Aufgaben

2.1 Liste mit Resultaten

Gesucht sind drei sortierte Ranglisten der drei Kategorien und eine Namen-Referenzliste die nach Namen sortiert ist und alle Laufzeiten enthaltet.

Resultate als Textdateien:

2.1.1 Aufbau der Ranglisten-Dateien (junioren.rl, senioren.rl, elite.rl)

Spaltentitel in der ersten Zeile: "Rang Start-Nr. Laufzeit Name". In den folgenden Zeilen sind die Läufer dieser Kategorie mit Ihren Resultaten aufgeführt (in der Rang-Reihenfolge: 1, 2, 3 ...)

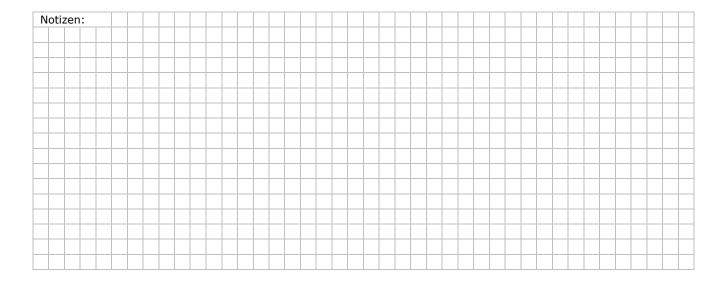
Das Felder-Trennzeichen ist der Tabulator

```
1 131 0:37:54 Gähwiler
2 049 1:38:17 Hugi
3 ...
```

2.1.2 Aufbau der Namen-Liste (namen.ref)

Spaltentitel in der ersten Zeile: "Startnummer Name Kategorie Rang Laufzeit". In den folgenden Zeilen sind die Läufer aller Kategorien mit Ihren Resultaten aufgeführt, sortiert nach Namen. Das Felder-Trennzeichen ist der Tabulator

Das i ciaci	Tremizerenen ibe der Tabalator			
118	Achermann	2	24	1:08:24
7	Achermann	1	16	0:38:01
61	Ammann	1	40	0:44:54
229	Ammann	3	33	1:23:56
174	Amschwand 1	2	4	1:02:09
175	• • •			

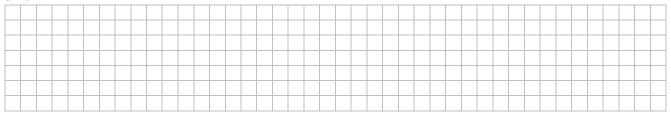




3 Aufgaben

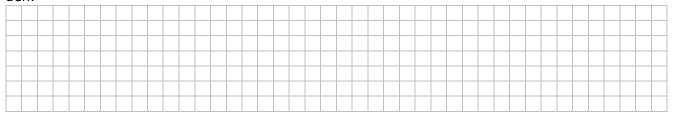
3.1 Datenanalyse

Führen Sie eine Datenanalyse durch, erstellen Sie den Plan der Datenstrukturen und Dateien resp. der entsprechenden Klassen. Hier sieht man den Ablauf der Daten von welchen Dateien diese kommen und wohin diese zwischengespeichert werden und danach welche Dateien entstanden sind!



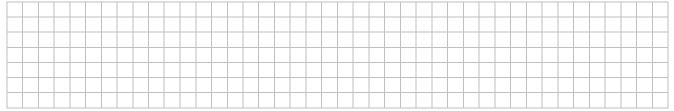
3.2 Klassen suchen und definieren

Analysieren Sie die Listen und die Aufgabenstellung und definieren Sie die nötigen Klassen welche für die Applikation sinnvoll erscheinen (Klassendiagramm). Es soll eine Konsolenapplikation sein und die Ausgabe soll sich auf ein minimum reduzieren, die nötigen Schritte sollen angezeigt werden.



3.3 Algorithmus

Überlegen Sie sich, wie Sie das Programm aufbauen werden und welche wesentlichen Schritte es beinhalten muss. Zeichnen Sie ein Ablaufplan als Flussdiagramm, als Aktivitätsdiagramm oder ein anderes Diagramm Ihrer Wahl, aus welchem der Ablauf des Programms ersichtlich ist (Prinzip Lösung).



4 Abgaben

4.1 Das Programm

Zippen Sie das Java Projekt und legen Sie es in den entsprechenden Abgabeordner.

M411_PRGSL_Name_Vorname.zip

4.2 Diagramme und Notizen

Speichern Sie auch die Diagramm und Notizen als PDF im entsprechenden Abgabeordner.

M411 DOKU Name Vorname.zip