

#67- Sortierung von Fußballspielern

Die Textdatei *Spieler.txt* enthält verschiedene Daten europäischer Fußballspieler. Es ist ein Programm zu erstellen mit dem die Spieler sortiert ausgegeben werden können.

Name	Spielposition	Alter	Verein	Liga	Sp	Tore	Assist
0123456789012345678901234567890123456789012345678901234567890123456789012345678							
Luis Suárez	Mittelstürmer	29	FC Barcelona	LaLiga	11	8	4
Edinson Cavani	Mittelstürmer	29	FC Paris SG	Ligue 1	11	11	0

- Definiere eine Klasse **Spieler** in der die Daten für jeden Spieler abgespeichert werden können.
- Lies die Spielerdaten ein und speichere sie in einer **List<Spieler>** ab.
- Programmiere ein Menü bei dem der Benutzer verschiedene sortierte Listen ausgeben kann:

1. Sortiere alle Spieler in aufsteigender alphabetischer Reihenfolge (A bis Z)

Auszugebene Spalten: Name, Verein, Geschossene Tore, Assist, Punkte

Punkte = Geschossene Tore + Assists

z. B. Messi hat in 9 Spielen, insgesamt 8 Tore geschossen und 3 Assist gegeben, d. h. er hat $8+3=11$ Punkte

2. Sortiere alle Spieler nach ihren geschossenen Toren (aufsteigend) und dem Nachnamen (aufsteigend)

Auszugebende Spalten: Name, Geschossene Tore

D. h. Wenn mehrere Spieler die gleiche Anzahl geschossener Tore haben, sollen diese nach ihrem Nachnamen sortiert werden.

3. Alle Spieler absteigend sortiert nach ihrer Trefferquote

(Trefferquote=Geschossene Tore/Anzahl der Spiele)

Auszugebende Spalten: Name, Trefferquote, Geschossene Tore, Anzahl der Spiele

Die Trefferquote soll auf zwei Nachkommastellen gerundet werden.

Hinweise:

- Es werden aus der Datei nur der Vor- und Nachname, die Spiele, die Tore und die Assist benötigt.

Zur Sortierung:

- Leite für die 1. Liste die Klasse **Spieler** von **Comparable<Spieler>** ab.
- Definiere für die 2. und 3. Listenausgabe zwei verschiedene Comparer Klassen und übergebe ein Objekt dieser Klassen bei Sort als Parameter.

Zur Ausgabe:

- Die Rundung einer Trefferquote `quote` auf zwei Nachkommastellen erfolgt am einfachsten durch die Funktion `Math.Round(quote, 2)`.
- Eine spaltenorientierte Ausgabe einer Liste ist leicht erreichbar, in dem man die Spaltenbreite im Format-String angibt (- bedeutet, dass die Ausgabe nach links ausgerichtet wird), z. B.
`Console.WriteLine("{0,-28}{1,4}{2,-20}", Vorname+" "+Nachname, Tore, Verein);`