

Soccer Table: Bewertungsraster

BBZW, Modul 426

Patrick Bucher

06.06.2024

Anforderungen

1. Funktionale

1. Der Anwender kann den Ligaordner der zu verarbeitenden Liga angeben; nur dieser wird für die Tabelle berücksichtigt.
2. Der Anwender kann den letzten zu berücksichtigenden Spieltag angeben; nur die Spielergebnisse bis zu diesem werden für die Tabelle berücksichtigt.
3. Die Spalten Rang, Name, Punkte, Anzahl Siege, Anzahl Niederlagen, Anzahl Unentschieden, erzielte Tore, kassierte Tore und Tordifferenz sind vorhanden und werden korrekt berechnet.
4. Die Sortierung der Zeilen erfolgt nach den folgenden Kriterien: 1) Anzahl Punkte (absteigend), 2) Tordifferenz (absteigend), 3) Anzahl Siege (absteigend) und 4) Name (aufsteigend).
5. Die Tabelle wird übersichtlich dargestellt und erlaubt das Herauskopieren von Text.
6. Spalten mit numerischen Werten werden rechtsbündig dargestellt. Alle Spalten haben eine passende Überschrift, wobei die Breite der Tabelle 120 Zeichen nicht übersteigen sollte.

2. Nicht-funktionale

1. Die Abgabe erfolgt pünktlich bis am **Montag, den 17.06.2024 um 8:00 Uhr** durch Bekanntgabe der URL eines Git-Repositories, auf das die Lehrperson Zugriffsberechtigungen hat, per E-Mail an die Adresse patrick.bucher@sluz.ch mit dem Betreff Modul 426: Abgabe Projekt Soccer Table [Klasse: Gruppenname], wobei Sie [Klasse: Gruppenname] durch Ihre Klasse bzw. Ihren Gruppennamen ersetzen.
 - Bei verspäteter Abgabe innerhalb der gleichen Woche erfolgt ein Notenabzug von bis zu 2.0 Notenpunkten.
 - Erfolgt die Abgabe mehr als eine Woche verspätet oder gar nicht, wird das Projekt mit der Note 1.0 bewertet.
2. Das Repository ist zum Abgabezeitpunkt mit dem Tag v1.0.0 versehen.

3. Ein README.md ist im root-Verzeichnis des Repositories vorhanden und vollständig, dies umfasst: 1) Instruktion zur Installation (Angabe der Voraussetzungen, Installation der Abhängigkeiten), 2) Instruktion zur Ausführung der Software, 3) Instruktionen zur Ausführung der Unittests. Die Instruktionen sollen für Windows oder Linux funktionieren.
 4. Die Kernlogik (Filterung der Spieltage, Berechnung der Spalten, Sortierung gemäss aller Kriterien) ist signifikant mithilfe von Unittests abgedeckt.
 5. Es werden zusätzliche Testdaten zur Verfügung gestellt, um alle vier Sortierkriterien testen zu können (manueller Test).
 6. Der Code ist unter den *Clean Code*-Aspekten Formatierung, Benennung, Kommentare, Wiederverwendbarkeit und Klarheit qualitativ hochwertig.
 7. Die Anwendung ist gemäss *Simple Design* so einfach wie möglich gehalten und weist keine unnötige (d.h. weiter reduzierbare) Komplexität auf.
3. Individueller Beitrag
1. Jedes Team erhält pro Teammitglied 10 Punkte zur freien internen Verteilung. Die Punkteverteilung wird teamintern vorgenommen und in der Datei contributions.md im root-Verzeichnis des Repositories **als Vorschlag** ausgewiesen (mit Zeilen im Format Name: Punktzahl). Die Lehrperson kann diese Verteilung übernehmen oder gemäss eigener Einschätzung anpassen, wobei Rücksprache mit der Gruppe gehalten wird.

Punkteverteilung

#	Punkte
1.1	2
1.2	3
1.3	8
1.4	4
1.5	3
1.6	4
2.1	1
2.2	1
2.3	4
2.4	8
2.5	2
2.6	5
2.7	3
3.1	10
Total	60

Testdaten

Die Testdaten stehen im Repository [m426/soccer-results](#) zur Verfügung. Zu Bewertungszwecken können weitere gleichartige Testdaten (gleiches Dateinamen- und Datenformat, Datenmengen innerhalb der gleichen Grössenordnung) herangezogen werden.