

# Einstieg in Java und OOP

Christian Silberbauer

## Übungsblatt 9

Nachfolgende Aufgaben beziehen sich auf die schrittweise Entwicklung eines einfachen „BundesligaManagers“ zur Verwaltung der Punktetabelle einer Spielsaison. Im Ergebnis soll sich die Oberfläche der Anwendung entsprechend der Abbildung darstellen.

Die einzelnen Spalten können nach ihrem Inhalt sortiert werden und die Punkte können durch unmittelbare Eingabe angepasst werden. Letzteres führt ggf. zur Änderung der Platzierung und zur automatischen Umsortierung der Tabelle.



Platz ▲	Club	Punkte
1	FC Bayern München	78
2	Borussia Dortmund	76
3	RasenBallsport Leipzig	66
4	Bayer 04 Leverkusen	58
5	Borussia M'gladbach	55
6	VfL Wolfsburg	55
7	Eintracht Frankfurt	54
8	SV Werder Bremen	53
9	TSG 1899 Hoffenheim	51
10	Fortuna 95 Düsseldorf	44
11	1. FSV Mainz 05	43
12	Hertha BSC	43
13	SC Freiburg	36
14	FC Schalke 04	33
15	FC Augsburg	32
16	VfB Stuttgart	28
17	Hannover 96	21
18	1. FC Nürnberg	19

### Aufgabe 1

Implementieren Sie eine Klasse `TabellenEintrag`, um einen Eintrag der Tabelle zu speichern. Die Klasse benötigt dazu folgende Attribute:

- `platz`: Platzierung
- `clubName`: Name des Vereins
- `punkte`: Aktueller Punktstand

Kapseln Sie bitte die Attribute, indem Sie entsprechende `get`- und `set`-Methoden zur Verfügung stellen und implementieren Sie zudem einen Initialisierungskonstruktor.

### Aufgabe 2

Implementieren Sie eine weitere Klasse `Tabelle`, die eine Liste von `TabellenEintrag`-Objekten verwaltet. Die Klasse `Tabelle` besitzt die Methoden:

- `void addEintrag(String clubname, int punkte)`: Platzierung ermitteln sowie ggf. der bisherigen Einträge ändern, `TabellenEintrag`-Instanz erzeugen und der Liste hinzufügen
- `TabellenEintrag getEintrag(int index)`: `TabellenEintrag`-Instanz an einem bestimmten Index in der Liste zurückgeben
- `void updatePunkte(int index, int punkte)`: Punkte eines Clubs an einem bestimmten Index anpassen und die Platzierungen der betroffenen Clubs ändern.

### Aufgabe 3

Erstellen Sie die Klasse `TabelleTableModel` als Model für die GUI. Die entsprechenden Daten werden in Form eines `Tabelle`-Objekts bereitgestellt. Die Zellen der Punkte-Spalte sollen editierbar sein. Nutzen Sie hierzu insbesondere die Methoden `isCellEditable()` und `setValueAt()`. Sie können zum Bearbeiten der Zelle den Standard-`CellEditor` verwenden und brauchen keine eigene Implementierung dessen bereitstellen.

#### Aufgabe 4

Erstellen Sie einen `JFrame` namens `BundesligaManager`, der die Anzeige der Anwendung realisiert. Er muss für die Darstellung der Tabelle ein `JTable`-Objekt verwenden, dessen Model eine Instanz von `TabelleTableModel` ist. Erlauben Sie die Sortierung der Spalten mittels `setAutoCreateRowSorter()`. Orientieren Sie sich beim Design bitte an der obigen Abbildung. Zum Test der Anwendung können Sie folgende `main()`-Methode verwenden:

```
public static void main(String[] args) {

    // Bundesligatabelle nach dem letzten Spieltag der Saison 2018/2019

    Tabelle tabelle = new Tabelle();

    tabelle.addEintrag("1. FC Nürnberg", 19);
    tabelle.addEintrag("1. FSV Mainz 05", 43);
    tabelle.addEintrag("Bayer 04 Leverkusen", 58);
    tabelle.addEintrag("Borussia Dortmund", 76);
    tabelle.addEintrag("Borussia M'gladbach", 55);
    tabelle.addEintrag("Eintracht Frankfurt", 54);
    tabelle.addEintrag("FC Augsburg", 32);
    tabelle.addEintrag("FC Bayern München", 78);
    tabelle.addEintrag("FC Schalke 04", 33);
    tabelle.addEintrag("Fortuna 95 Düsseldorf", 44);
    tabelle.addEintrag("Hertha BSC", 43);
    tabelle.addEintrag("Hannover 96", 21);
    tabelle.addEintrag("RasenBallSport Leipzig", 66);
    tabelle.addEintrag("SC Freiburg", 36);
    tabelle.addEintrag("SV Werder Bremen", 53);
    tabelle.addEintrag("TSG 1899 Hoffenheim", 51);
    tabelle.addEintrag("VfB Stuttgart", 28);
    tabelle.addEintrag("VfL Wolfsburg", 55);

    new BundesligaManager(tabelle);
}
```