

46116 Frühjahr 2018

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Thema Nr. 1	3
Teilaufgabe Nr. 1	3
Aufgabe 3 (Objektorientierte Implementierung) [Fußballweltmeisterschaft]	3



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



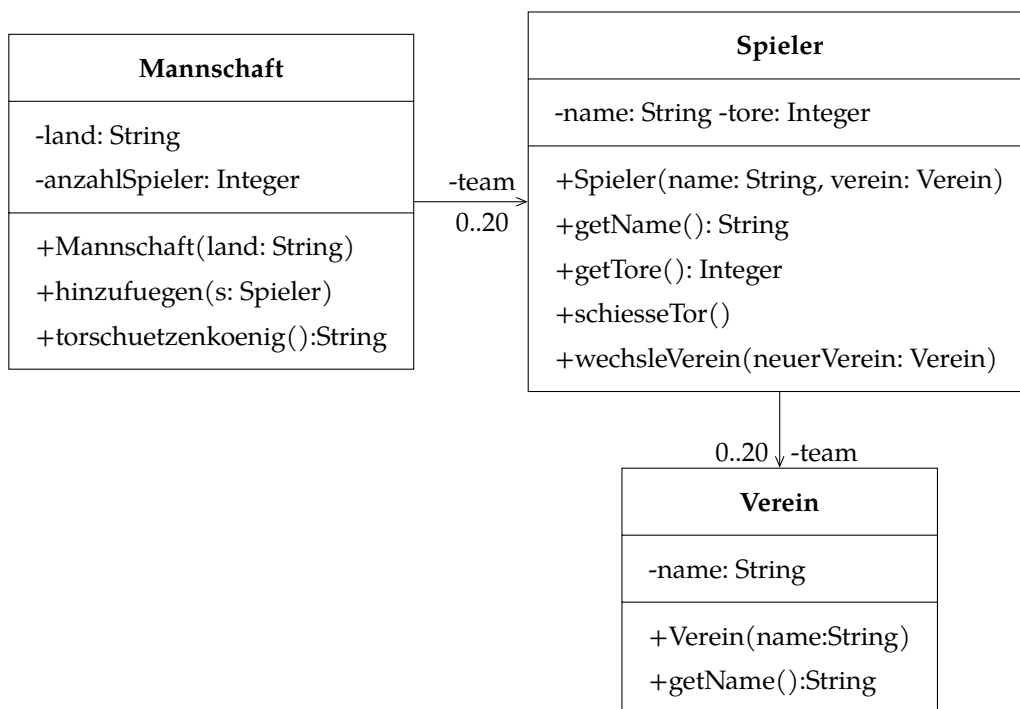
Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Thema Nr. 1

Teilaufgabe Nr. 1

Aufgabe 3 (Objektorientierte Implementierung) [Fußballweltmeisterschaft]

Für die nächste Fußballweltmeisterschaft möchte ein Wettbüro ein Programm zur Verwaltung von Spielern, Vereinen und (National-)Mannschaften entwickeln. Dazu wurde bereits das folgende UML-Klassendiagramm entworfen.



Es kann angenommen werden, dass die Klasse **Verein** bereits implementiert ist. In den folgenden Implementierungsaufgaben können Sie eine objektorientierte Programmiersprache Ihrer Wahl verwenden. Die verwendete Sprache ist anzugeben. Zu beachten sind jeweils die im Klassendiagramm angegebenen Sichtbarkeiten von Attributen, Rollennamen, Konstruktoren und Operationen.

- (a) Es ist eine Implementierung der Klasse **Spieler** anzugeben. Der Konstruktor soll die Instanzvariablen mit den gegebenen Parametern initialisieren, wobei die Anzahl der Tore nach der Objekterzeugung gleich 0 sein soll. Ansonsten kann die Funktionalität der einzelnen Operationen aus deren Namen geschlossen werden.

Lösungsvorschlag

```
public class Spieler {
    private String name;
    private int tore;
    @SuppressWarnings("unused")
    private Verein verein;

    public Spieler(String name, Verein verein) {
```

```

    this.name = name;
    this.verein = verein;
    tore = 0;
}

public String getName() {
    return name;
}

public int getTore() {
    return tore;
}

public void schiesseTor() {
    tore++;
}

public void wechsleVerein(Verein neuerVerein) {
    verein = neuerVerein;
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Spieler.java](https://github.com/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Spieler.java)

- (b) Es ist eine Implementierung der Klasse `Mannschaft` anzugeben. Der Konstruktor soll das Land initialisieren, die Anzahl der Spieler auf 0 setzen und das Team mit einem noch „leeren“ Array der Länge 20 initialisieren. Das Team soll mit der Methode `hin_zufuegen` um einen Spieler erweitert werden. Die Methode `torschuetzenkoenig` soll den Namen eines Spielers aus dem Team zurückgeben, der die meisten Tore für die Mannschaft geschossen hat. Ist das für mehrere Spieler der Fall, dann kann der Name eines beliebigen solchen Spielers zurückgegeben werden. Ist noch kein Spieler im Team, dann soll der String `"Kein Spieler vorhanden"` zurückgegeben werden.

Lösungsvorschlag

```

public class Mannschaft {
    @SuppressWarnings("unused")
    private String land;
    private int anzahlSpieler;
    private Spieler[] team;

    public Mannschaft(String land) {
        this.land = land;
        anzahlSpieler = 0;
        team = new Spieler[20];
    }

    public void hinzufuegen(Spieler s) {
        team[anzahlSpieler] = s;
        anzahlSpieler++;
    }

    public String torschuetzenkoenig() {
        if (anzahlSpieler == 0) {

```

```

        return "Kein Spieler vorhanden";
    }
    Spieler koenig = team[0];
    for (int i = 0; i < anzahlSpieler; i++) {
        Spieler spieler = team[i];
        if (spieler.getTore() > koenig.getTore()) {
            koenig = spieler;
        }
    }
    return koenig.getName();
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Mannschaft.java](https://github.com/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Mannschaft.java)

- (c) Schreiben Sie den Rumpf einer `main`-Methode, so dass nach Ausführung der Methode eine deutsche Mannschaft existiert mit zwei Spielern Namens `"Hugo Meier"` und `"Frank Huber"`. Beide Spieler sollen zum selben Verein `"FC Staatsexamen"` gehören. `"Hugo Meier"` soll nach Aufnahme in die deutsche Mannschaft genau ein Tor geschossen haben, während `"Frank Huber"` noch kein Tor erzielt hat. (Wir abstrahieren hier von der Realität, in der ein 2-er Team noch gar nicht spielbereit ist.)

Lösungsvorschlag

```

public class Verein {
    private String name;

    public Verein(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public static void main(String[] args) {
        Verein verein1 = new Verein("FC Staatsexamen");
        Spieler spieler1 = new Spieler("Hugo Meier", verein1);
        Spieler spieler2 = new Spieler("Frank Huber", verein1);
        Mannschaft deutscheMannschaft = new Mannschaft("Deutschland");
        deutscheMannschaft.hinzufuegen(spieler1);
        deutscheMannschaft.hinzufuegen(spieler2);
        spieler1.schiesseTor();
    }
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Verein.java](https://github.com/bschlangaul/examen/examen_46116/jahr_2018/fruehjahr/verein/Verein.java)