

Lösungsvorschlag: Notenberechnung

TODO 1

```
// TODO 1: Erstellen Sie eine Klasse "Note" mit folgenden Attributten:  
//           Notenwert (1.0 -6.0), Datum, Gewichtung  
  
// TODO 1.1: Erstellen Sie sinnvolle Konstruktoren  
  
// TODO 1.2: Erstellen Sie sinnvolle getter/setter  
  
// TODO 1.3: Implementieren Sie die Methode toString();  
  
// TODO 1.4: Comitten und Pushen Sie  
  
import java.time.LocalDate;  
  
/**  
 * Eine Schulnote mit Wert, Gewichtung und Datum  
 * @author Kevin Maurizi  
 * @since 2042.01.01  
 * @version 0.1  
 */  
public class Note {  
    private double notenwert;  
    private double gewichtung;  
    private LocalDate datum;  
  
    public Note(double notenwert, double gewichtung, LocalDate datum) {  
        setNotenwert(notenwert);  
        this.gewichtung = gewichtung;  
        this.datum = datum;  
    }  
  
    public double getNotenwert() {  
        return notenwert;  
    }  
  
    public void setNotenwert(double notenwert) {  
        if(notenwert < 1.0 || notenwert > 6.0){  
            System.err.println("Die Note muss zwischen 1.0 und 6.0 liegen.");  
        }  
        this.notenwert = notenwert;  
    }  
}
```

```
public double getGewichtung() {
    return gewichtung;
}

public void setGewichtung(double gewichtung) {
    this.gewichtung = gewichtung;
}

public LocalDate getDatum() {
    return datum;
}

public void setDatum(LocalDate datum) {
    this.datum = datum;
}

@Override
public String toString() {
    return "Note{" +
        "notenwert=" + notenwert +
        ", gewichtung=" + gewichtung +
        ", datum=" + datum +
        '}';
}
}
```



Die `setNotenwert(double notenwert)` Methode beinhaltet eine Validierung (`notenwert < 1.0 || notenwert > 6.0`). Deshalb wird im Konstruktor die Methode aufgerufen um den Wert zu speichern und nicht `this.notenwert = notenwert`. Damit lässt sich doppelter Code vermeiden und auch beim Erstellen der Note im Konstruktor wird die eingabe validiert.

TODO 2

```
// TODO 2: Erstellen Sie die Klasse "Fach" mit folgenden Attributen
//           Liste von Noten, Name

// TODO 2.1: Erstellen Sie sinnvolle Konstruktoren

// TODO 2.2: Erstellen Sie sinnvolle getter/setter/adder
```

```
// TODO 2.3: Implementieren Sie die Methode toString();  
  
// TODO 2.4: Comitten und Pushen Sie  
  
  
import java.util.ArrayList;  
  
/**  
 * Ein Schulfach mit einem Namen und einer Liste von Noten.  
 * @author Kevin Maurizi  
 * @since 2042.01.01  
 * @version 0.1  
 */  
public class Fach {  
  
    private String name;  
    private ArrayList<Note> noten;  
  
    public Fach(String name) {  
        this.name = name;  
        noten = new ArrayList<>();  
    }  
  
    public void addNote(Note note){  
        noten.add(note);  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return "Fach{" +  
            "name='" + name + '\'' +  
            ", noten=" + noten +  
            ", durchschnitt=" + berechneDurchschnitt() +  
            '}';  
    }  
}
```

TODO 3

```
// TODO 3: Erstellen Sie die Methode "berechneDurchschnitt()" die den Durchschnitt der Noten berechnet
//           unter berücksichtigung der Gewichtung!

/**
 * Checkt ob die Summe der Gewichtungen 100% ergibt und gibt dann den Notenschnitt aus, wenn die Gewichtung
 * nicht 100% ergibt erscheint eine Fehlermeldung und der Durchschnitt ist -1.
 * @return Notenschnitt aus allen Noten in der ArrayList
 */
public double berechneDurchschnitt(){
    double durchschnitt = 0;
    double gewichtungen = 0;
    for (Note note : noten) {
        gewichtungen = gewichtungen + note.getGewichtung();
        durchschnitt = durchschnitt +
note.getGewichtung()*note.getNotenwert();
    }

    if(gewichtungen != 1){
        System.err.println("Die Gewichtungen müssen in der Summe 100% ergeben!");
        durchschnitt = -1;
    }

    return durchschnitt;
}
```

TODO 4 & 5

```
private void run() {
    // TODO 4: Erstellen Sie 2 Fächer mit jeweils 2-4 Noten und überprüfen Sie, ob alles korrekt funktioniert
    // (Erstellen, Durschnitt berechnen etc.).

    Fach fach1 = new Fach("Mathe");
    Note note1 = new Note(5.5,0.5, LocalDate.of(2021,12,12));
    fach1.addNote(note1);
    fach1.addNote( new Note(5.0,0.5, LocalDate.of(2021,12,12)));
    System.out.println(fach1.berechneDurchschnitt());
```

```
Fach fach2 = new Fach("Deutsch");
fach2.addNote( new Note(4.8,0.25, LocalDate.of(2021,12,12)));
fach2.addNote( new Note(4.8,0.6, LocalDate.of(2021,12,12)));
fach2.addNote( new Note(4.8,0.1, LocalDate.of(2021,12,12)));
System.out.println(fach2.berechneDurchschnitt());

// TODO 5: Drucken Sie alle Fächer und Noten aus, verwenden Sie dazu
ihre toString()-Methode
System.out.println(fach1);
System.out.println(fach2);
}
```

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki



Permanent link:

<https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu08/loesungen/notenberechnung>

Last update: **2024/03/28 14:07**