

Lösungsvorschlag: Java und die Zeit

TODO 1 - 4

```
// TODO 1: Erstellen Sie eine Klasse Person mit folgenden
Attributen:
//   Nachname, Vorname, Adresse (Strasse, Ort, PLZ), Geburtsdatum
//   Nebenbemerkung:
//       1. Adresse ist eine eigene Klasse
//       2. Geburtsdatum hat den Datentyp "LocalDate"

// TODO 2: Erstellen Sie sinnvolle Konstruktoren

// TODO 3: Erstellen Sie sinnvolle getter/setter

// TODO 4: Implementieren Sie die Methode toString();

import java.time.LocalDate;
import java.time.Period;

/**
 * Einfache Klasse um eine Person zu speichern.
 */
public class Person{
    private String vorname, nachname;
    private Adresse adresse;
    private LocalDate geburtsdatum;

    public Person(String vorname, String nachname, Adresse adresse,
LocalDate geburtsdatum) {
        this.vorname = vorname;
        this.nachname = nachname;
        this.adresse = adresse;
        this.geburtsdatum = geburtsdatum;
    }

    public Person(String vorname, String nachname, LocalDate geburtsdatum,
String strasse, String ort, int plz) {
        this.vorname = vorname;
        this.nachname = nachname;
        this.adresse = new Adresse(strasse, ort, plz);
        this.geburtsdatum = geburtsdatum;
    }

    public Person(String vorname, String nachname, int geburtsjahr, int
geburtsmonat, int geburtsstag, String strasse, String ort, int plz) {
        this.vorname = vorname;
```

```
        this.nachname = nachname;
        this.adresse = new Adresse(strasse, ort, plz);
        this.geburtsdatum =
LocalDate.of(geburtsjahr, geburtsmonat, geburtstag);
    }

    public String getVorname() {
        return vorname;
    }

    public String getNachname() {
        return nachname;
    }

    public Adresse getAdresse() {
        return adresse;
    }

    public LocalDate getGeburtsdatum() {
        return geburtsdatum;
    }

    public void setVorname(String vorname) {
        this.vorname = vorname;
    }

    public void setNachname(String nachname) {
        this.nachname = nachname;
    }

    public void setAdresse(Adresse adresse) {
        this.adresse = adresse;
    }

    @Override
    public String toString() {
        return "Person{" +
            "vorname='" + vorname + '\'' +
            ", nachname='" + nachname + '\'' +
            ", adresse=" + adresse +
            ", geburtsdatum=" + geburtsdatum +
            '}';
    }
}

/**
 * Einfache Klasse für eine Adresse
```

```
*/  
public class Adresse {  
    private String strasse;  
    private String ort;  
    private int plz;  
  
    public Adresse(String strasse, String ort, int plz) {  
        this.strasse = strasse;  
        this.ort = ort;  
        this.plz = plz;  
    }  
  
    @Override  
    public String toString() {  
        return "Adresse{" +  
            "strasse='" + strasse + '\'' +  
            ", ort='" + ort + '\'' +  
            ", plz=" + plz +  
            '}';  
    }  
  
    public String getStrasse() {  
        return strasse;  
    }  
  
    public void setStrasse(String strasse) {  
        this.strasse = strasse;  
    }  
  
    public String getOrt() {  
        return ort;  
    }  
  
    public void setOrt(String ort) {  
        this.ort = ort;  
    }  
  
    public int getPlz() {  
        return plz;  
    }  
  
    public void setPlz(int plz) {  
        this.plz = plz;  
    }  
}
```

TODO 6 & 7

// TODO 6: Erstellen Sie die Methode getAlter(), die das Alter aus dem Geburtsdatum berechnet.

// TODO 7: Machen Sie die Klasse Person vergleichbar nach Alter (Implementieren Sie Comparable<T>)

```
import java.time.LocalDate;
import java.time.Period;

/**
 * Einfache Klasse um eine Person zu speichern.
 */
public class Person implements Comparable<Person>{
    private String vorname, nachname;
    private Adresse adresse;
    private LocalDate geburtsdatum;

    public Person(String vorname, String nachname, Adresse adresse,
LocalDate geburtsdatum) {
        this.vorname = vorname;
        this.nachname = nachname;
        this.adresse = adresse;
        this.geburtsdatum = geburtsdatum;
    }

    public Person(String vorname, String nachname, LocalDate geburtsdatum,
String strasse, String ort, int plz) {
        this.vorname = vorname;
        this.nachname = nachname;
        this.adresse = new Adresse(strasse, ort, plz);
        this.geburtsdatum = geburtsdatum;
    }

    public Person(String vorname, String nachname, int geburtsjahr, int
geburtsmonat, int geburtstag, String strasse, String ort, int plz) {
        this.vorname = vorname;
        this.nachname = nachname;
        this.adresse = new Adresse(strasse, ort, plz);
        this.geburtsdatum =
LocalDate.of(geburtsjahr, geburtsmonat, geburtstag);
    }

    public String getVorname() {
        return vorname;
    }
}
```

```
public String getNachname() {
    return nachname;
}

public Adresse getAdresse() {
    return adresse;
}

public LocalDate getGeburtsdatum() {
    return geburtsdatum;
}

public void setVorname(String vorname) {
    this.vorname = vorname;
}

public void setNachname(String nachname) {
    this.nachname = nachname;
}

public void setAdresse(Adresse adresse) {
    this.adresse = adresse;
}

public int getAlter(){
    LocalDate jetzt = LocalDate.now();
    Period alter = Period.between(geburtsdatum,jetzt);
    return alter.getYears();
}

@Override
public String toString() {
    return "Person{" +
        "vorname='" + vorname + '\'' +
        ", nachname='" + nachname + '\'' +
        ", adresse=" + adresse +
        ", geburtsdatum=" + geburtsdatum +
        '}';
}

@Override
public int compareTo(Person o) {
    return getAlter()-o.getAlter();
}
}
```



```
@Override
public int compareTo(Person o) {
    return getAlter()-o.getAlter();
}
```



Was passiert hier? Wir vergleichen das Alter von zwei Personen. Ist die eine Person älter wie die Person o so muss eine positive Zahl zurückgegeben werden. Ist also das alter 34 und o.getAlter() ist 30 ist das Resultat 4, also eine positive Zahl. Genau so funktioniert es auch anders herum, versuchen Sie es aus!

TODO 8 & 9 & 10

```
// TODO 8: Erstellen Sie 5 Personen und fügen Sie diese einer Liste hinzu.
ArrayList<Person> personenListe = new ArrayList<>();
personenListe.add(new Person("Kevin", "Maurizi", 1989, 7, 5, "Meinradsweg 3", "Grabs", 9472));
personenListe.add(new Person("Anna", "Maurizi", 2019, 6, 7, "Meinradsweg 3", "Grabs", 9472));
personenListe.add(new Person("Emma", "Maurizi", 2019, 6, 7, "Meinradsweg 3", "Grabs", 9472));
personenListe.add(new
Person("Angela", "Maurizi", 1984, 2, 21, "Meinradsweg 3", "Grabs", 9472));
personenListe.add(new Person("Peter", "Müller", 1954, 1, 23, "Dorfbachweg 4", "Zürich", 8000));

// TODO 9: Lassen Sie die Liste sortieren.
Collections.sort(personenListe);

// TODO 10: Geben Sie die Liste aus.
for (Person person : personenListe) {
    System.out.println(person);
}
```

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

Permanent link:
<https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu08/loesungen/javaunddiezeit>

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

