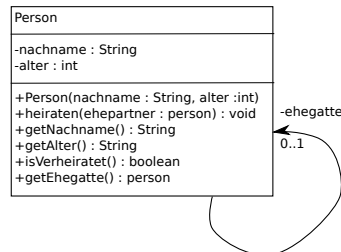


## 8. Übungsblatt - Informatik 1 - Lösungsbeispiele

Die Klasse Person sei analog zur Vorlesung wie folgt gegeben:



### Aufgabe 1 (JUnit)

Die Methode `scheiden()` trennt eine Person von ihrem Ehepartner. Falls diese Person einen Ehepartner hatte, dann haben nach Methodenaufruf weder dieser Ehepartner noch diese Person sich gegenseitig als Ehepartner: sie sind wieder ledig. Vor-, Nachname und Alter bleiben unverändert. Falls die Person, bei der `scheiden()` aufgerufen wird, kein Ehepartner hat, dann macht `scheiden()` nichts.

Implementieren Sie eine JUnit-Testklasse, mit der Sie die Methode `scheiden()` testen. Mit Ihren Tests müssen alle beschriebenen Funktionen der Methode `scheiden()` überprüft werden.

#### Lösungsvorschlag:

Es werden mindestens zwei Testmethoden benötigt: eine für den verheiratet Fall und eine andere für unverheiratet.

```
public class PersonTest extends TestCase {

    public void testScheiden1 () {
        Person meier = new Person("Meier", 18);
        Person mayer = new Person("Mayer", 77);

        meier.scheiden ();

        assertFalse(meier.isVerheiratet ());
        assertFalse(mayer.isVerheiratet ());
        assertEquals("Meier", meier.getNachname ());
        assertEquals("Mayer", mayer.getNachname ());
        assertEquals(18, meier.getAlter ());
        assertEquals(77, mayer.getAlter ());
    }

    public void testScheiden2 () {
        Person meier = new Person("Meier", 18);
```

```

    Person mayer = new Person("Mayer", 77);

    meier.heiraten(mayer);
    meier.scheiden();

    assertEquals("Meier-Mayer", meier.getNachname());
    assertEquals("Meier-Mayer", mayer.getNachname());
    assertEquals(18, meier.getAlter());
    assertEquals(77, mayer.getAlter());
    assertEquals(mayer, meier.getEhegatte());
    assertEquals(meier, mayer.getEhegatte());
}
}

```

## Aufgabe 2 (MVC (View))

Entwerfen und implementieren Sie einen Sicht, die eine Objektmethode enthält, welche den Vor-, Nachname und das Alter von der Tastatur einliest und eine Person mit den eingegebenen Werten zurück gibt. Geben Sie den Entwurf als UML-Klassendiagramm an.

Verwenden Sie die Klasse Scanner, um Werte einzulesen:

- Ein Scanner-Object mit `new Scanner(System.in)` erzeugen.
- mit den Objektmethoden `nextLine()` bzw. `nextInt()` String-Werte bzw. int-Werte von der Konsole einlesen.

Der Benutzerdialog der Eingabe soll wie folgt aussehen:

Bitte geben Sie ihre persönlichen Daten ein.

Nachname: Meier

Alter: 77

Fehlerfälle brauchen nicht behandelt zu werden.

Lösungsvorschlag:

```

import java.util.Scanner;

public class PersonTestEingabe {

    public Person eingeben() {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Bitte geben Sie ihre persönlichen Daten ein.");
    }
}

```

```

        System.out.print("Nachname:␣");
        String nachname = scanner.nextLine();
        System.out.print("Alter:␣");
        int alter = scanner.nextInt();

        return new Person(nachname, alter);
    }
}

```

### Aufgabe 3 (MVC (Controller))

Entwerfen und implementieren Sie eine Steuerung, die den Benutzer solange auffordert die Daten für eine Person einzugeben, bis eine volljährige Person eingegeben wurde. Sie können die Klasse aus der vorangehenden Aufgabe zur Eingabe der Personendaten verwenden. Falls Sie bei den anderen vorhandenen Klassen wie Person oder der Klasse aus der vorangehenden Aufgabe noch Methoden hinzufügen, dann geben Sie diese Methoden vollständig an.

Lösungsvorschlag:

```

public class VolljaehrigePersonSteuerung {

    public static void main(String[] args) {
        VolljaehrigePersonSteuerung steuerung
            = new VolljaehrigePersonSteuerung();

        steuerung.start();
    }

    public void start() {
        Person volljaehrigePerson;
        PersonTextEingabe eingabe = new PersonTextEingabe();

        do {
            volljaehrigePerson = eingabe.eingeben();
        } while (! volljaehrigePerson.isVolljaehrig());
    }
}

```

Die Überprüfung auf Volljährigkeit gehört zum Modell:

```

public class Person {

    .....
}

```

```
public boolean isVolljaehrig() {  
    return this.alter >= 18;  
}  
}
```