

## Adressbuch, 1. Ausbaustufe

Verwendet: Klassen, Attribute, Methodenaufrufe, Konstruktoren.

Zunächst soll ein ganz simples Adressbuch erstellt werden, welches sich lediglich ein Person-Exemplar merken und dieses wieder zurückliefern kann.

### Die Klasse Person:

Wir benötigen zunächst eine Klasse `Person`. Objekte dieser Klasse sollen zwei Attribute vom Typ `String` mit den Bezeichnern `firstName` und `lastName` besitzen. Außerdem soll `Person` eine Methode `getPersonAsString()` besitzen, welche einen `String` liefert, der aus den Werten von `firstName` und `lastName` besteht, mit einem Leerzeichen dazwischen. Damit wir `Person`-Exemplare sinnvoll initialisieren können, soll `Person` einen Konstruktor haben, dem man zwei `String`s als Parameter übergeben kann, mit denen dann die beiden Attribute belegt werden.

### Die Klasse AddressBook:

In der ersten, allereinfachsten Form soll ein Exemplar dieser Klasse genau ein `Person`-Exemplar aufnehmen können. Dieses legen wir daher in `AddressBook` in einem Attribut vom Typ `Person` mit dem Bezeichner `person` ab. Außerdem bekommt die Klasse zwei Methoden:

1. Eine Methode `setPerson()`. Diese hat keinen Rückgabewert, und akzeptiert als Parameter ein `Person`-Exemplar. Die Methode soll das übergebene Exemplar in das Attribut `person` schreiben.
2. Eine Methode `getPerson()`. Diese liefert das in "person" enthaltene `Person`-Exemplar zurück.

Mit den obigen Vorgaben können wir nun bereits eine Testklasse schreiben, *bevor* wir die beiden Klassen implementieren:

```
package de.mpaap.addressbook;

public class AddressBookTest {
    public static void main(String[] args) {

        // create person
        String firstName = "Michael";
        String lastName = "Paap";
        Person p1 = new Person(firstName, lastName);

        // create addressbook and fill in person
        AddressBook book = new AddressBook();
        book.setPerson(p1);

        // fetch person from book and print it
        Person p2 = book.getPerson();
        System.out.println(p2.getPersonAsString());
    }
}
```

### Aufgabe:

Schreiben Sie die Klassen `Person` und `AddressBook` gemäß der obigen Vorgaben und testen Sie sie mit Hilfe der Klasse `AddressBookTest`.