

Hinweis: Es sind alle Felder auszufüllen! Abgabe der Übungsblätter immer **mittwochs** (Ausnahme wenn Feiertag: donnerstags) bis **spätestens 12:00 Uhr** in die entsprechend gekennzeichneten Briefkästen der Veranstaltung im Erdgeschoss des Instituts für Informatik (Gebäude N). Zuwiderhandlung wird mit Strafe geahndet! (Punktabzug)

Übungsblatt	
-------------	--

(hier die Nummer des bearbeiteten **Übungsblatts** eintragen)

	Übung 01 (1057 N) Montag 08:15 - 09:45 Uhr (Isabell Rücker)
	Übung 02 (1056 N) Montag 14:00 - 15:30 Uhr (Henning Cui)
	Übung 03 (1057 N) Montag 15:45 - 17:15 Uhr (Josef Kircher)
	Übung 04 (1054 N) Montag 17:30 - 19:00 Uhr (Mosaab Slimani)
	Übung 05 (1057 N) Montag 17:30 - 19:00 Uhr (David Hacker)
	Übung 06 (1055 N) Dienstag 12:15 - 13:45 Uhr (André Schweiger)
X	Übung 07 (1054 N) Dienstag 17:30 - 19:00 Uhr (Benjamin Sertolli)
	Übung 08 (1057 N) Dienstag 17:30 - 19:00 Uhr (Dat Le Thanh)
	Übung 09 (1054 N) Mittwoch 08:15 - 09:45 Uhr (Erik Pallas)
	Übung 10 (1055 N) Mittwoch 08:15 - 09:45 Uhr (Moritz Feldmann)
	Übung 11 (1054 N) Mittwoch 10:00 - 11:30 Uhr (Denise Böhm)
	Übung 12 (1056 N) Donnerstag 08:15 - 09:45 Uhr (Florian Magg)
	Übung 13 (1054 N) Donnerstag 15:45 - 17:15 Uhr (Marvin Drexelius)
	Übung 14 (1054 N) Donnerstag 17:30 - 19:00 Uhr (Patrick Eckert)
	Übung 15 (1057 N) Donnerstag 17:30 - 19:00 Uhr (Alexander Szöke)
	Übung 16 (1057 N) Freitag 08:15 - 09:45 Uhr (Philipp Braml)
	Übung 17 (1054 N) Freitag 10:00 - 11:30 Uhr (Elisabeth Korndörfer)
	Übung 18 (1054 N) Freitag 12:15 - 13:45 Uhr (Philipp Häusele)
	Übung 19 (1056 N) Freitag 12:15 - 13:45 Uhr (Maximilian Demmler)
	Übung 20 (1054 N) Freitag 14:00 - 15:30 Uhr (Florian Straßer)

(hier die eingeteilte **Übungsgruppe** ankreuzen)

Teamnummer	6
------------	---

(hier die Nummer des eingeteilten **Teams** eintragen)

Tarik Selimovic
Anton Lydike
Dominic Cesnak

(hier die **Vor- und Nachnamen** aller Teammitglieder eintragen)

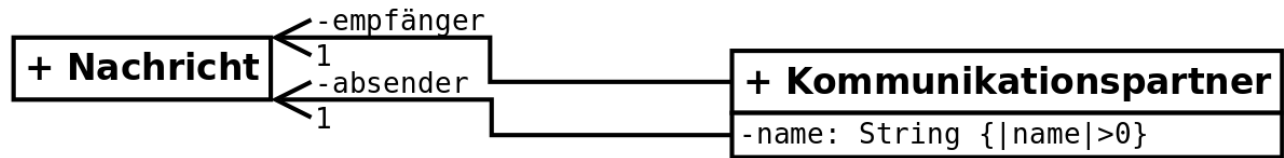
Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
<b>Gesamt</b>		

(vom Tutor auszufüllen)

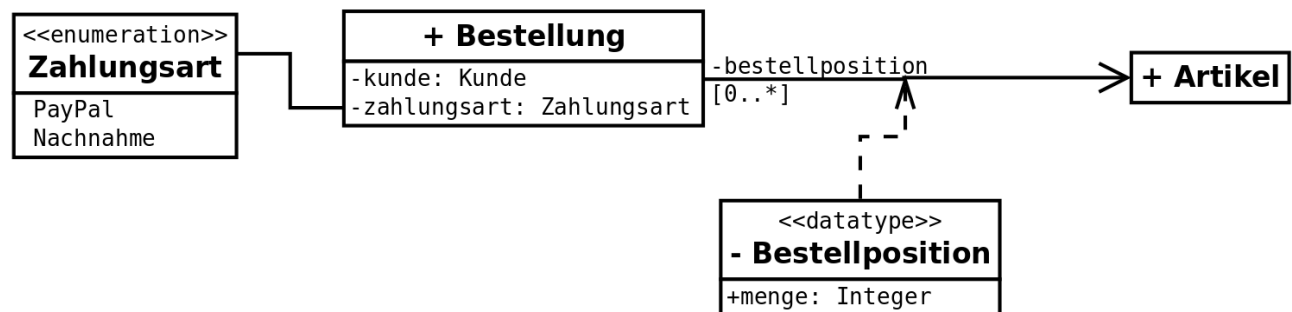
# Übungsblatt 4

13)

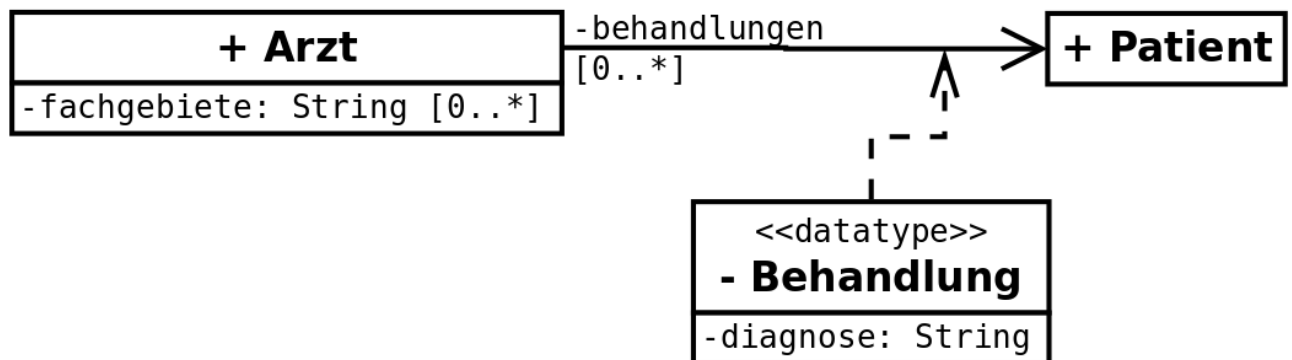
a)



b)

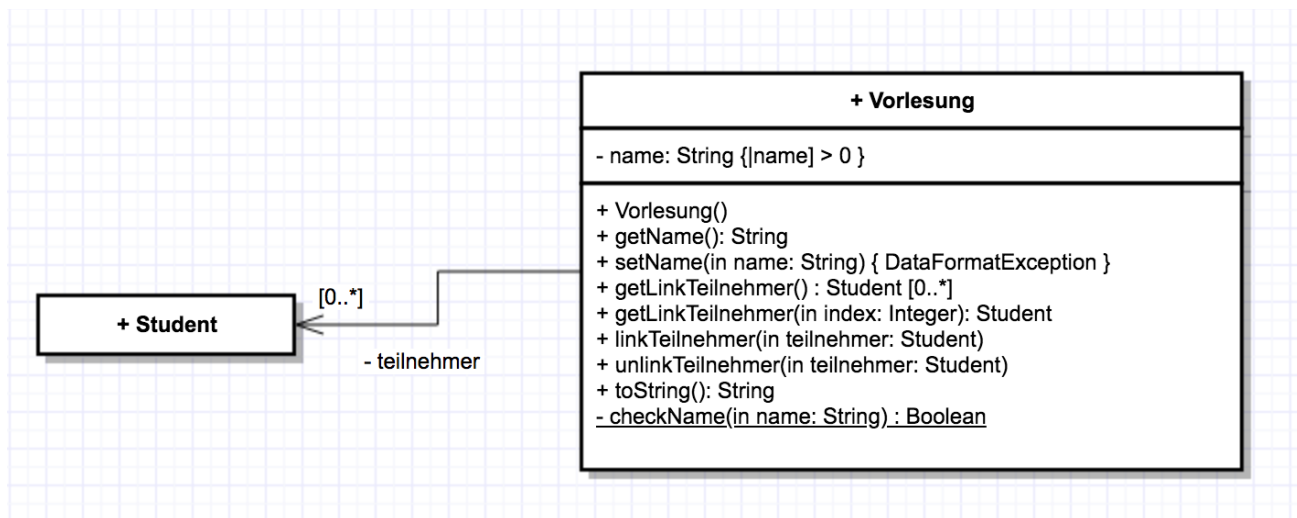


c)

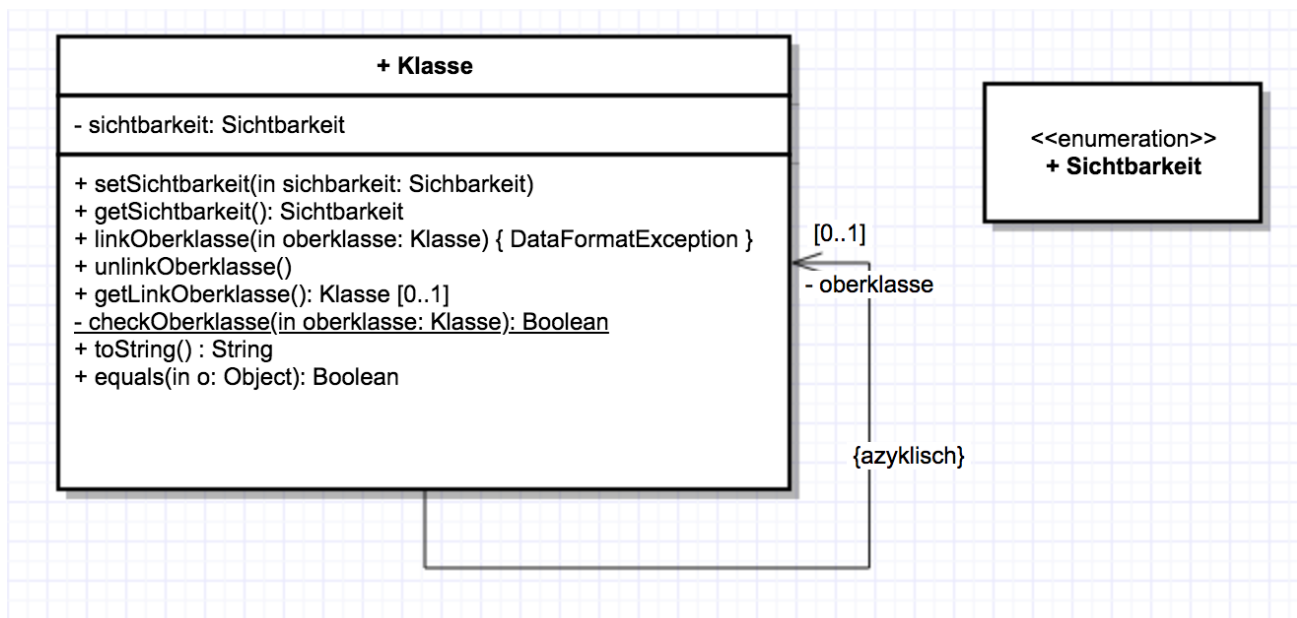


14)

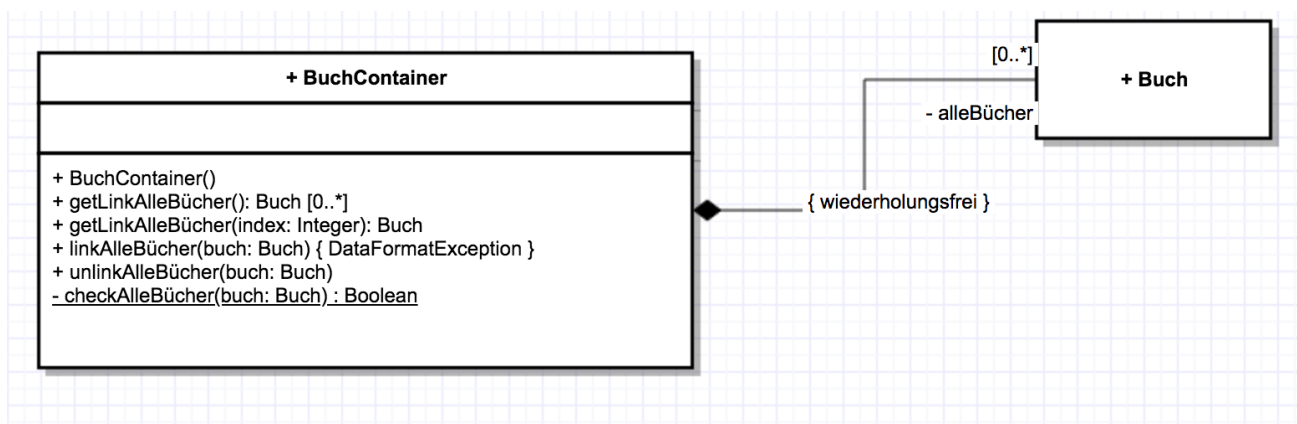
a)



b)



c)



## 15a)

---

### Student)

```
package aufgabe15a;

import java.util.ArrayList;
import java.util.zip.DataFormatException;

public class Student {
    private ArrayList<Studiengang> studiengänge;

    public Student (Studiengang s) throws DataFormatException {
        studiengänge = new ArrayList<Studiengang>();

        linkStudiengang(s);
    }

    public void linkStudiengang(Studiengang s) throws DataFormatException {
        if (checkStudiengang(s)) {
            studiengänge.add(s);
        } else {
            throw new DataFormatException("Studiengang ist bereits enthalten.");
        }
    }

    private boolean checkStudiengang(Studiengang s) {
        return studiengänge.contains(s);
    }
}
```

### Studiengang)

```
package aufgabe15a;

public class Studiengang {

}
```

## 15b)

### Student)

```
package aufgabe15b;

import java.util.ArrayList;
import java.util.zip.DataFormatException;

public class Student {
    private ArrayList<Studium> studiumse;

    public Student (Studium s) throws DataFormatException {
        studiumse = new ArrayList<Studium>();

        linkStudium(s);
    }

    public void linkStudium(Studium s) throws DataFormatException {
        if (checkStudium(s)) {
            studiumse.add(s);
        } else {
            throw new DataFormatException("Studium ist bereits enthalten.");
        }
    }

    private boolean checkStudium(Studium s) {
        return studiumse.contains(s);
    }
}
```

### Studiengang)

```
package aufgabe15b;

public class Studiengang {

}
```

### Studium)

```
package aufgabe15b;

import java.util.Date;

public class Studium {
    private Date immatrikulation;
    private Date exmatrikulation;
    private Studiengang studiengang;

    public Studium (Date im) {
        setImmatrikulation(im);
    }

    public void setImmatrikulation(Date im) {
        immatrikulation = im;
    }
}
```

## 16a)

---

### Evaluation)

```
package aufgabe16a;

import java.util.Date;

public class Evaluation {
    private double grade;
    private String remark;
    private int attempt;
    private Date date;
    private Student examinee;

    public Evaluation (Student s) {
        setExaminee(s);
    }

    public void setExaminee(Student s) {
        examinee = s;
    }
}
```

### Exam)

```
package aufgabe16a;

import java.util.ArrayList;

public class Exam {
    private ArrayList<Evaluation> evaluations;

    public void linkEvaluation (Evaluation e) {
        evaluations.add(e);
    }

    public void unlinkEvaluation(Evaluation e) {
        evaluations.remove(e);
    }
}
```

### Student)

```
package aufgabe16a;

public class Student {

}
```

## 16b)

### Directory)

```
package aufgabe16b;

import java.util.ArrayList;
import java.util.zip.DataFormatException;

public class Directory {
    private ArrayList<File> fileelements;
    private ArrayList<Directory> directoryelements;

    public ArrayList<File> getLinkFileelement() {
        return fileelements;
    }

    public void linkFileelement(File f) throws DataFormatException{
        if (checkFileelement(f)) {
            fileelements.add(f);
        } else {
            throw new DataFormatException("Folder already contains that file.");
        }
    }

    private boolean checkFileelement(File f) {
        return !fileelements.contains(f);
    }

    public void unlinkFileelement(File f) {
        fileelements.remove(f);
    }

    public ArrayList<Directory> getlinkDirectoryelement() {
        return directoryelements;
    }

    public void linkDirectoryelement(Directory d) throws DataFormatException{
        if (checkDirectoryelement(d)) {
            directoryelements.add(d);
        } else {
            throw new DataFormatException("Folder already contains that folder.");
        }
    }

    private boolean checkDirectoryelement(Directory d) {
        return !directoryelements.contains(d);
    }

    public void unlinkDirectoryelement(Directory d) {
        directoryelements.remove(d);
    }
}
```

### File)

```
package aufgabe16b;

public class File {

}
```

