

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)

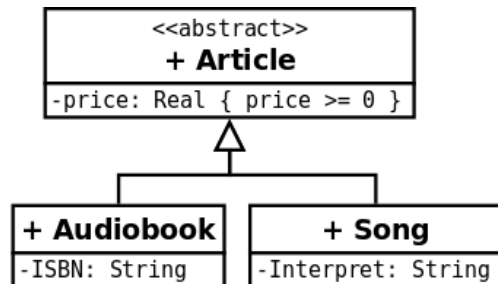
Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
Gesamt		

(vom Tutor auszufüllen)

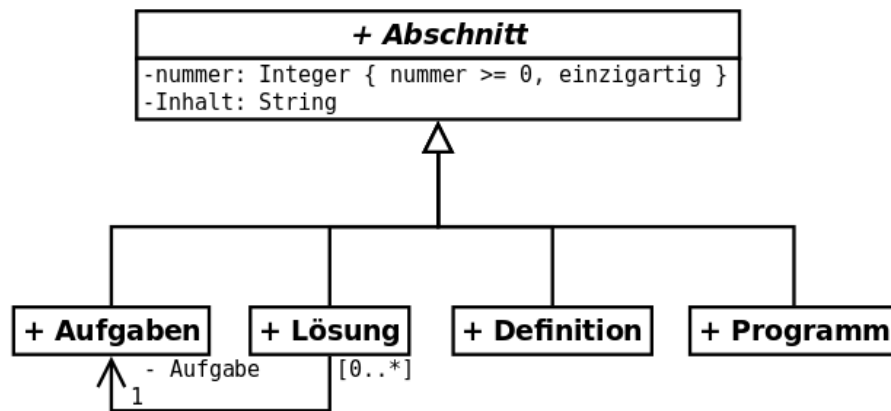
Übungsblatt 5

17)

a)

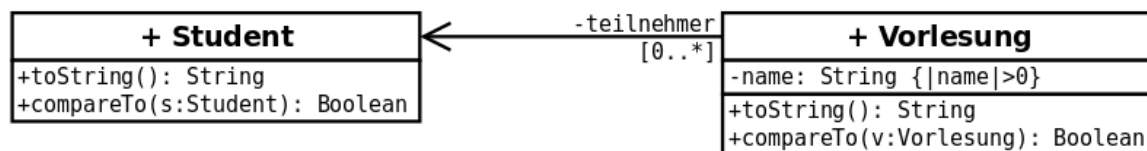


b)

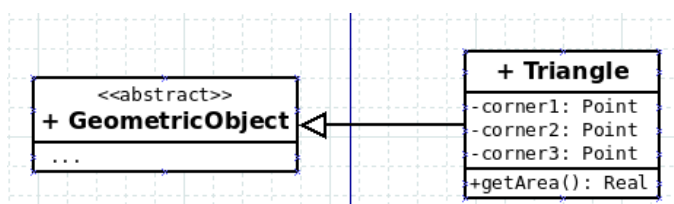


18)

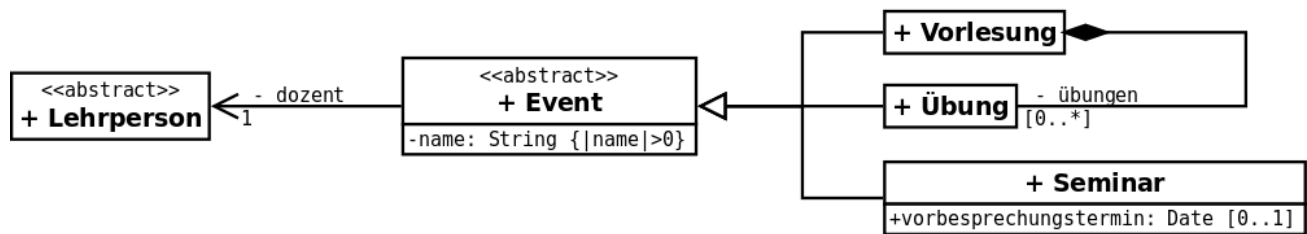
a)



b)



c)



19a)

Grundstück)

```

package aufgabe19a;

public class Grundstück extends Immobilie {
    private int gröÙe;

    public Grundstück (double preis, int gröÙe) {
        super(preis);

        this.gröÙe = gröÙe;
    }

    public int getGröÙe() {
        return gröÙe;
    }

    public String toString() {
        return "Wohnung: {"
            + "\n\tpreis: " + getPreis()
            + "\n\tgröÙe: " + getGröÙe()
            + "\n}";
    }
}

```

Immobilie)

```

package aufgabe19a;

public abstract class Immobilie {
    private double preis;

    protected Immobilie (double preis) {
        this.preis = preis;
    }

    public double getPreis() {
        return preis;
    }
}

```

Wohnung)

```
package aufgabe19a;

public class Wohnung extends Immobilie {
    private int wohnfläche;

    public Wohnung(double preis, int wohnfläche) {
        super(preis);

        this.wohnfläche = wohnfläche;
    }

    public int getWohnfläche() {
        return wohnfläche;
    }

    public String toString() {
        return "Wohnung: {"
            + "\n\tpreis: " + getPreis()
            + "\n\tfläche: " + getWohnfläche()
            + "\n}";
    }
}
```

19b)

Kugel)

```
package aufgabe19b;

public class Kugel extends Körper {
    private int radius;

    public Kugel (int radius) {
        super();
        this.radius = radius;
    }

    public double getVolumen() {
        return 4/3 * Math.PI * radius * radius * radius;
    }

    public int getRadius () {
        return radius;
    }

    public boolean equals(Object o) {
        return (o instanceof Kugel && ((Kugel) o).getRadius() == radius);
    }
}
```

Körper)

```
package aufgabe19b;

public abstract class Körper {
    public abstract double getVolumen();
}
```

Quader)

```

package aufgabe19b;

public class Quader extends Körper {
    private int länge;
    private int breite;
    private int höhe;

    public Quader(int länge, int breite, int höhe) {
        super();
        this.länge = länge;
        this.breite = breite;
        this.höhe = höhe;
    }

    public double getVolumen() {
        return länge * breite * höhe;
    }

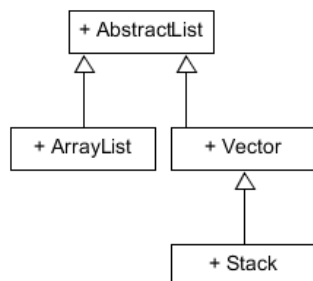
    public int getLänge() {
        return länge;
    }

    public int getBreite() {
        return breite;
    }

    public int getHöhe() {
        return höhe;
    }

    public boolean equals(Object o) {
        return (o instanceof Quader
            && ((Quader) o).getLänge() == länge
            && ((Quader) o).getBreite() == breite
            && ((Quader) o).getHöhe() == höhe);
    }
}

```

20)**a)**

b)

