Universität Augsburg
Institut für Informatik

Deckblatt: Übung zur Vorlesung Informatik 2 SS 2018
Fakultät für Angewandte Informatik
Lehrprofessur für Informatik
Prof. Dr. Lorenz, Marius Brendle, Johannes Metzger

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

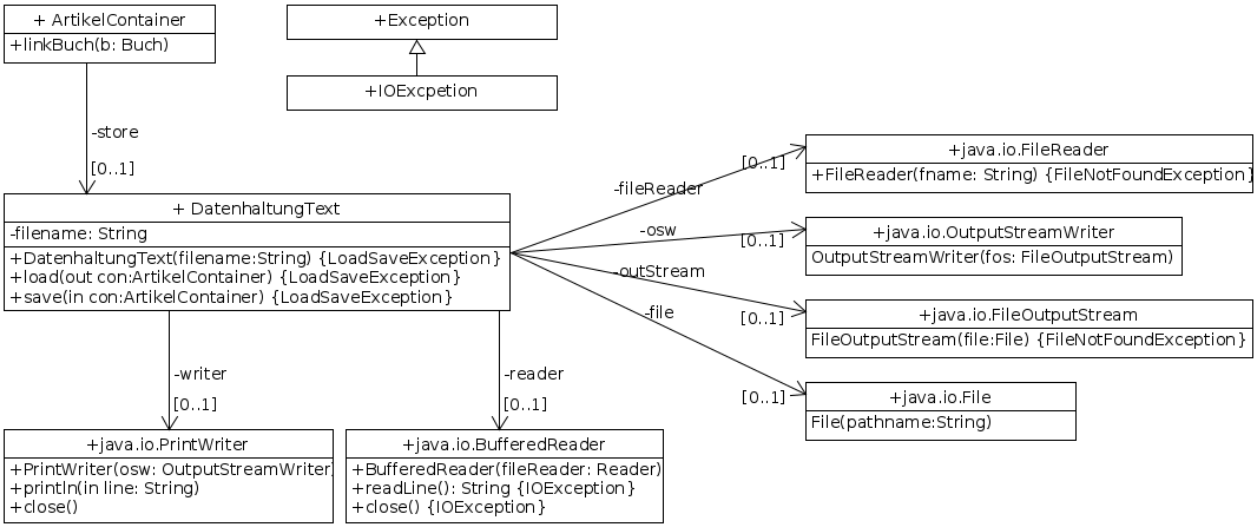
(hier die **Vor- und Nachnamen** aller Teammitglieder **eintragen**)

Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
Gesamt		(vom Tutor auszufüllen)

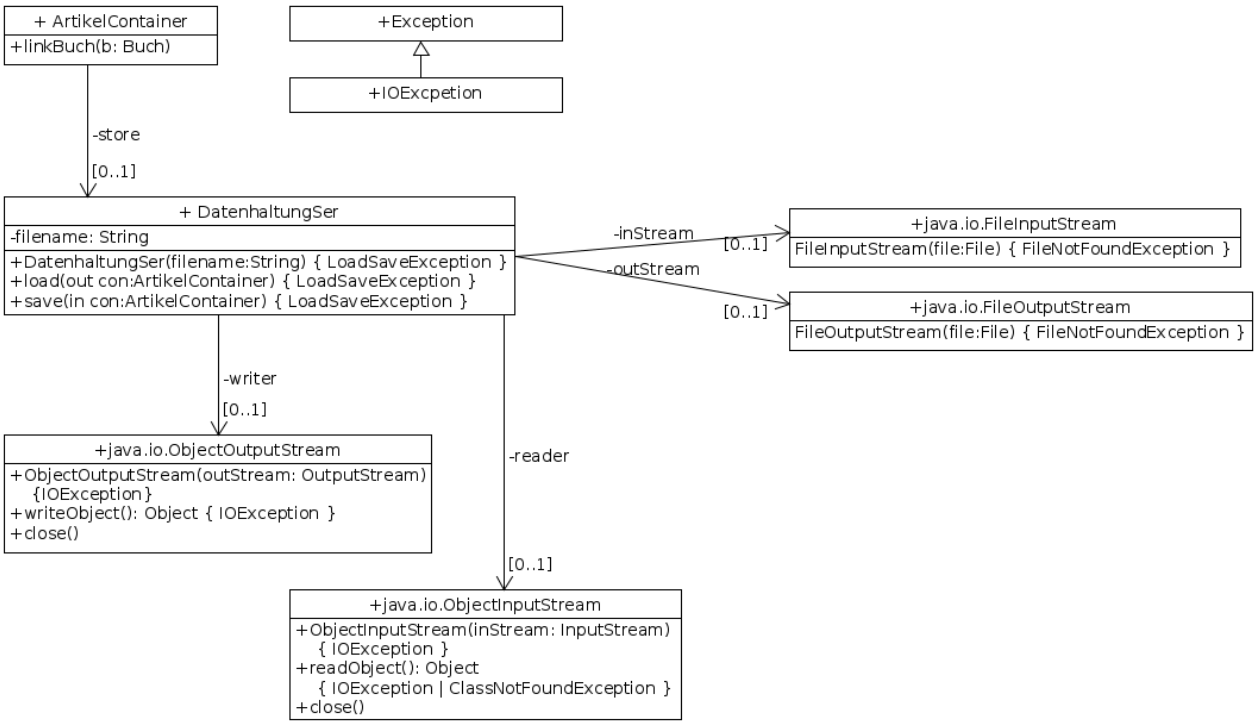
Übungsblatt 11

41)

a)



b)



42)

Objektdatei)

```
package aufgabe42;

import java.io.*;
import java.util.Iterator;

import aufgabe43.*;

public class Objektdatei {

    public String filename;

    public Objektdatei(String filename) throws PersistenceException {
        if (filename == null)
            throw new PersistenceException("Filename shan't be null! (But this excpetions cause shall
be)", null);
        this.filename = filename;
    }

    public void load (ArtikelContainer con) throws PersistenceException {
        ArtikelContainer tmp;
        try (ObjectInputStream reader = new ObjectInputStream(new FileInputStream(filename))) {
            tmp = (ArtikelContainer) reader.readObject();
            for (Artikel a: tmp) {
                con.linkArtiel(a);
            }
        } catch (IOException | ClassNotFoundException ex) {
            throw new PersistenceException("Reading failed", ex);
        }
    }

    public void save (ArtikelContainer con) throws PersistenceException {
        try (ObjectOutputStream writer = new ObjectOutputStream(new FileOutputStream(filename))) {
            writer.writeObject(con);
        } catch (IOException ex) {
            throw new PersistenceException("Saving failed", ex);
        }
    }
}
```

43)

Textdatei)

```
package aufgabe43;

import java.io.*;
import java.util.Iterator;

public class Textdatei {

    private BufferedReader reader;
    private PrintWriter writer;
    private String filename;

    public Textdatei(String filename) throws PersistenceException {
        if (filename == null) throw new PersistenceException("Filename is null", null);
        this.filename = filename;
    }

    public void load(BuchContainer con) throws PersistenceException {
        try {
            reader = new BufferedReader(new FileReader(this.filename));
            String isbn, titel, line = reader.readLine();

            while (!line.equals("end")) {
                if (!line.equals("new")) {
                    reader.close();
                    throw new PersistenceException("Invalid format!", null);
                }
                try {
                    isbn = reader.readLine();
                    titel = reader.readLine();
                    if (titel == null) {
                        // muss nicht werfen, macht aber sinn, da Datei anscheinend beschädigt.
                        throw new PersistenceException("Invalid end-of-input", null);
                    }
                    con.linkBuch(new Buch(isbn, titel));
                } catch(IOException e) {
                    throw new PersistenceException("Error while reading book store!", e);
                }
                line = reader.readLine();
            }
        } catch(IOException e) {
            throw new PersistenceException("Error while opening book store", e);
        } finally {
            try { reader.close(); } catch(Exception ee) {}
        }
    }

    public void save(BuchContainer con) throws PersistenceException {
        try {
            writer = new PrintWriter(new OutputStreamWriter(new FileOutputStream(new File(filename))));
            for (Buch b: con) {
                writer.println("new");
                writer.println(b.getISBN());
                writer.println(b.getTitel());
            }
            writer.println("end");
        } catch (IOException e) {
            throw new PersistenceException("Error while writing to book store!", e);
        } finally {
            writer.close();
        }
    }

}
```

44)

a)

```
package aufgabe44.a;

import blatt10.MausCanvas;

import java.awt.*;
import java.awt.event.*;

public class MausFrame extends Frame implements MouseListener {

    MausCanvas canvas;
    private int clickCounter = 0;
    private Label clickLabel = new Label();

    public MausFrame() {
        super("Maus Frame");

        setLayout(new FlowLayout());

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });

        addMouseListener(this);
        clickLabel.addMouseListener(this);
        add(clickLabel);

        setSize(100, 100);

        setVisible(true);
        clickLabel.setVisible(true);

    }

    public static void main(String args[]) {
        new MausFrame();
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        clickCounter = clickCounter +1;

        clickLabel.setText(String.valueOf(clickCounter));
        clickLabel.revalidate();

        clickLabel.setLocation(e.getX(), e.getY());
    }

    @Override
    public void mousePressed(MouseEvent e) { }

    @Override
    public void mouseReleased(MouseEvent e) { }

    @Override
    public void mouseEntered(MouseEvent e) { }

    @Override
    public void mouseExited(MouseEvent e) { }
}
```

b)

```
package aufgabe44.b;

import blatt10.MausCanvas;

import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class MausFrame extends Frame implements MouseListener {

    MausCanvas canvas;

    private Label clickLabel = new Label();

    public MausFrame() {
        super("Maus Frame");
        setLayout(new FlowLayout());

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
                System.exit(0);
            }
        });

        addMouseListener(this);
        clickLabel.addMouseListener(this);
        add(clickLabel);

        setSize(100, 100);


        setVisible(true);
        clickLabel.setVisible(true);

    }

    public static void main(String args[]) {
        new MausFrame();
    }

    @Override
    public void mouseClicked(MouseEvent e) {

        clickLabel.setText("clicked");
        clickLabel.revalidate();

        clickLabel.setLocation(e.getX(), e.getY());
    }

    @Override
    public void mousePressed(MouseEvent e) {
        clickLabel.setText("pressed");
        clickLabel.revalidate();
        clickLabel.setLocation(e.getX(), e.getY());

    }

    @Override
    public void mouseReleased(MouseEvent e) {

        clickLabel.setText("clicked");
        clickLabel.revalidate();

        clickLabel.setLocation(e.getX(), e.getY());
```

```
    }

    @Override
    public void mouseEntered(MouseEvent e) { }

    @Override
    public void mouseExited(MouseEvent e) { }
}
```