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

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

Aufgabe		
Aufgabe		
Aufgabe		
Aufgabe		
Gesamt		(vom Tutor auszufüllen)

Übungsblatt 2

5)

a)

```
package aufgabe5;

public class A {
  public static void main(String[] args) {
    if (args.length != 2
      || !args[0].equals("-o")
      || !Character.isDigit(args[1].charAt(0))
     ) {
      System.out.println("nicht ok");
    } else {
      System.out.println("ok");
    }
  }
}
```

b)

```
package aufgabe5;

public class B {
  public static void main(String[] args) {
    StringBuilder s = new StringBuilder();

    for (int i = 0; i < args.length; i++) {
      s.append(args[i]);
    }

    System.out.println("Länge: " + s.length());
  }
}
```

6)

a)

```
package aufgabe6;

public class A {

  public static void main(String[] args) {
    if (args.length != 2) {
      System.err.println("Not enogh numbers!");
      return;
    }

    try {
      double a = Double.parseDouble(args[0]);
      double b = Double.parseDouble(args[1]);

      System.out.println("Ergebnis: " + (a * b));
    } catch (NumberFormatException e) {
      System.err.println("Couldn't convert!");
    }
  }

}
```

b)

```
package aufgabe6;

import java.util.ArrayList;

public class B {

  public static void main(String[] args) {
    ArrayList<Double> list = new ArrayList<Double>();
    int n = randomIntRange(1, 100000);
    double sum = 0;

    while (n-- > 0) {
      list.add(Math.random());
    }

    n = list.size();

    while (n-- > 0) {
      sum += list.get(n);
    }

    System.out.println("Average: " + (sum / list.size()));
  }

  private static int randomIntRange (int min, int max) {
    max++;
    return (int) Math.floor(Math.random() * (max - min)) + min;
  }


}
```

7)

a)

```
/*
    i)

    0 < priority && priority < 6
    1 < |name| && |name| < 21
*/

package aufgabe7;

import java.util.zip.DataFormatException;
import vorgaben.Hobby;

public class A {
  public static void main(String[] args) {
    Hobby h;

    try {
      h = new Hobby("Lesen");
      h.setPriority(h.getPriority() + 1);
    } catch (DataFormatException e) {
      System.err.println("Error encountered while managing hobby: "
                + e.toString());
      e.printStackTrace();
    }
  }
}
```

b)

```
/*
    i)

    rowDimension > 0
    columnDimension > 0
*/

package aufgabe7;

import java.util.zip.DataFormatException;
import vorgaben.Matrix;

public class B {
  public static void main(String[] args) {
    Matrix m;

    try {
      m = new Matrix(2,3);
    } catch (DataFormatException e) {
      System.err.println("Couldn't handle matrix: " + e.toString());
      e.printStackTrace();
      return;
    }

    for (int i = 0; i < 2; i++) {
      for (int j = 0; j < 3; j++) {
        m.setValue(i, j, 1);
      }
    }
  }
}
```

8)

Property)

```
package aufgabe8;

import java.util.zip.DataFormatException;

public class Property {
  private double price;
  private int size;

  private static double fee = 3.45;

  public Property (double price, int size) throws DataFormatException {
    setPrice(price);
    setSize(size);
  }

  public void setPrice(double price) throws DataFormatException {
    if (!checkPrice(price)) {
      throw new DataFormatException("Invalid Price");
    }
    this.price = price;
  }

  public double getPrice() {
    return this.price;
  }

  private boolean checkPrice(double price) {
    return price > 0;
  }

  public void setSize(int size) throws DataFormatException {
    if (!checkSize(size)) {
      throw new DataFormatException("Invalid size");
    }
    this.size = size;
  }

  public int getSize() {
    return this.size;
  }

  private boolean checkSize(int size) {
    return size > 0;
  }

  public static double getFee () {
    return Property.fee;
  }
}
```