

## Aufgaben zur Lehrveranstaltung Laborpraktikum Software SMSB3300 / SMIB3300

### Aufgabenblatt

- A.1 Zum „Warmwerden“ (Online-Hilfe, Package)
- A.2 Hausaufgabe (Arrays)
- A.3 Hausaufgabe (Operatoren |, ||, &, &&)
- A.4 Hausaufgabe (String und StringBuilder)
- A.5 Hausaufgabe (Destruktor und finalize())
- A.6 Präsenzaufgabe (Klassen, Objekte)
- A.7 Präsenzaufgabe (Begriffe)
- A.8 Präsenzaufgabe (Abstrakte Klassen, Polymorphie)
- A.9 Hausaufgabe (Polymorphie)
- A.10 Hausaufgabe (ToString)
- A.11 Hausaufgabe (Klassen, abstrakte Klassen, Properties, String-Operationen, Polymorphie)

**Präsenzaufgaben** werden **im Labor** bearbeitet. Die Arbeit im Labor ist für Sie die Gelegenheit, Verständnisfragen so weit zu klären, dass Sie in der Lage sind, die nachfolgenden Hausaufgaben selbständig zu erledigen. Präsenzaufgaben werden nicht benotet, ihre Bearbeitung ist jedoch zum Verständnis wichtig. Präsenzaufgaben müssen von Ihnen bearbeitet werden, **bevor** Sie sich an die jeweils nachfolgenden Hausaufgaben machen.

Halten Sie sich also am besten einfach an die Reihenfolge der Aufgaben des Aufgabenblattes: zuerst A.1 bis A.5 zu Hause, dann im Labor A.6 bis A.8, anschließend A.9 bis A.11. Die Bearbeitung der Präsenzaufgaben weisen Sie direkt im Labor dem Betreuer nach (Vorführung). Falls Sie nicht fertig geworden sind, vereinbaren Sie mit dem Betreuer das weitere Vorgehen (z. B. Abgabe zusammen mit den Hausaufgaben). Die Präsenzaufgaben dieses Aufgabenblattes werden im Labor bearbeitet.

**Hausaufgaben** werden (je nach individuellem Tempo) im Labor begonnen und zu Hause fortgeführt. Hausaufgaben müssen Sie zu einem vorgegebenen Termin abgeben. Abgaben erfolgen in elektronischer Form. Hausaufgaben sind Gegenstand des Testats und der Benotung. Die Hausaufgaben dieses Aufgabenblattes sind **bis 07.10.2019 (8:00 Uhr) abzugeben (siehe Zeitplan)**. Verspätete Abgaben werden nicht berücksichtigt!

## Hinweise bei Problemen mit der Ausführung von Programmen direkt vom USB-Stick.

Zum Arbeiten mit IntelliJ oder Eclipse auf verschiedenen Rechnern empfiehlt es sich, den Workspace direkt auf einem USB-Stick abzuspeichern. Sie können einen lokal erstellten Workspace einfach über ihr Dateisystem (z.B. Microsoft Explorer oder Midnight Commander unter Linux) auf ihren USB-Stick und von dort auf einen zweiten Rechner kopieren. Sie können den Workspace auch direkt von Ihrem USB-Stick öffnen.

Sollten Sie Dateien aus einem anderen Workspace bzw. Projekt importieren wollen, so können Sie im Paket-Explorer mit der rechten Maustaste auf Ihr aktuelles Projekt klicken und dann den Menüpunkt „Importieren...“ wählen.

## Aufgabenblatt

### A.1 Zum „Warmwerden“ (Online-Hilfe, Package)

Machen Sie sich mit der Entwicklungsumgebung IntelliJ vertraut. Lesen Sie hierzu die unter <https://www.torsten-horn.de/techdocs/IntelliJ-IDEA.html> verfügbare Kurzeinführung. Wo können Sie Informationen nachschlagen? Wo finden Sie die Java-Sprachreferenz? Studieren Sie dort die Bedeutung des `package`-Schlüsselwortes.

## XOR

Machen Sie sich mit der Entwicklungsumgebung ECLIPSE vertraut. Lesen Sie hierzu die unter <https://www.java-tutorial.org/eclipse.html> verfügbare Kurzeinführung. Wo können Sie Informationen nachschlagen? Wo finden Sie die Java-Sprachreferenz? Studieren Sie dort die Bedeutung des `package`-Schlüsselwortes.

### A.2 Hausaufgabe (Arrays)

[2 Punkte]

Schreiben Sie ein Programm, das eine Zahl  $n$  ( $>0$ ) einliest und dann alle Quadratwurzeln der Zahlen von 1 bis  $n-1$  in einem Array abspeichert. Berechnen Sie anschließend aus diesem Array den Mittelwert dieser Wurzeln.

Hinweise: Verwenden Sie `Integer.parseInt`. Beachten Sie, dass es bei Division durch 0 zu Fehlern kommen kann.

### A.3 Hausaufgabe (Operatoren |, ||, &, &&)

[0,5 Punkte]

Für die logische Und- bzw. Oder-Verknüpfung gibt es je zwei Operatoren: `&` und `&&` bzw. `|` und `||`. Studieren Sie die Online-Hilfe zu Java und geben Sie alle Unterschiede mit eigenen Worten sowie einem Programmier-Beispiel wieder.

### A.4 Hausaufgabe (String, StringBuffer und StringBuilder)

[0,5 Punkte]

Lesen Sie im Online-Buch „Java ist auch eine Insel“ von Christian Ullenboom Kapitel 4. Probieren Sie die dort beschriebenen Beispiele selbst aus. Beschreiben Sie mit eigenen Worten (möglichst mit einem Beispiel), wann welche der Klassen `String`, `StringBuffer` und `StringBuilder` verwendet werden sollten, sowie deren Vor- bzw. Nachteile.

## A.5 Hausaufgabe (Destruktor und finalize())

[0,5 Punkte]

Lesen Sie im Online-Buch „Java ist auch eine Insel“ von Christian Ullenboom die Kapitel 1.2.10, 3.4.2., 5.5.6 und 8.3.7. Beschreiben Sie mit eigenen Worten die Funktionsweise des **GarbageCollector** und die Freigabe nicht benötigter Objekte. Beschäftigen Sie sich dabei auch mit den Konzepten eines Destruktors bzw. dessen Realisierung in Java sowie mit der Funktion *finalize()*.

## A.6 Präsenzaufgabe (Klassen, Objekte)

Implementieren Sie eine Klasse „Kraftfahrzeug“ mit einem Attribut „modell“ vom Typ Zeichenkette und einem weiteren Attribut „verbrauchProKilometer“ zur Speicherung einer Fließkommazahl.

Implementieren Sie folgenden Konstruktor:

```
public Kraftfahrzeug(String _modell, double _verbrauchProKilometer)
{
    // Ihr Code hier
}
```

Implementieren Sie sichtbare Zugriffsoperationen für den Zugriff auf das Feld „modell“.

Implementieren Sie eine Operation, die den Verbrauch für eine als Parameter übergebene Strecke berechnet.

```
public double verbrauch(int kilometer)
{
    // Ihr Code hier
}
```

Schreiben Sie folgendes Hauptprogramm:

```
public static void main(String[] args) {
    Kraftfahrzeug[] autoArr = new Kraftfahrzeug[2];
    //Der Focus verbraucht 6,5 Liter auf 100 km:
    autoArr[0] = new Kraftfahrzeug("Focus", 0.065);
    //Der Tesla verbraucht 0 Liter auf 100 km:
    autoArr[1] = new Kraftfahrzeug("Tesla", 0.00);

    int km = 400;
    System.out.printf("Verbrauch auf %d km:%n", km);
    for (int i = 0; i < autoArr.length; i++) {
        System.out.printf("%s: %.0f Liter %n",
            autoArr[i].getModell(),
            autoArr[i].verbrauch(km));
    }
}
```

## A.7 Präsenzaufgabe (Begriffe)

Verbinden Sie die Elemente Ihres Programms auf der linken Seite mit je einer Linie mit einem Begriff auf der rechten Seite. Achten Sie genau auf die Groß/Kleinschreibung der links stehenden Elemente.

1	<code>autoArr</code>	Klasse	A
2	<code>Kraftfahrzeug</code>	Sichtbarkeit	B
3	<code>autoArr[0]</code>	Zugriffsoperation	C
4	<code>verbrauch</code>	Kommentar	D
5	<code>getModell</code>	Array-Index	E
6	<code>verbrauchProKilometer</code>	Attribut	F
7	<code>km</code>	Operation	G
8	<code>// Der Tesla verbraucht ...</code>	Referenz auf ein Objekt	H
9	<code>_modell</code>	Parameter	I
10	<code>public</code>	lokale Variable	J
11	<code>[1]</code>	Array	K

## A.8 Präsenzaufgabe (Abstrakte Klassen, Polymorphie)

Implementieren Sie in Ihrer Klasse „Kraftfahrzeug“ eine Methode „fahre“ ohne Parameter und ohne Rückgabewert. Diese Operation soll einfach „Gas geben“ auf der Konsole ausgeben.

Implementieren Sie eine Basisklasse „Fahrzeug“ von Kraftfahrzeug. Auch in der Basisklasse soll es eine Methode „fahre“ geben. Diesmal jedoch als abstrakte Methode.

Implementieren Sie eine Klasse „Fahrrad“ als zweite Subklasse von Fahrzeug. Implementieren Sie in Fahrrad ebenfalls eine Methode „fahre“. Diese soll den Text „Antreten“ auf der Konsole ausgeben. Erweitern Sie Ihr Hauptprogramm wie folgt:

```
Fahrzeug[] fahrzeugArr = new Fahrzeug[2];
fahrzeugArr[0] = new Kraftfahrzeug ("Golf", 0.065);
fahrzeugArr[1] = new Fahrrad();
for (int j = 0; j < 2; j++)
{
    fahrzeugArr[j].fahre();
}
```

## A.9 Hausaufgabe (Polymorphie)

[1 Punkt]

Implementieren Sie in der Klasse `Fahrrad` ein Feld „größteZoll“ und sichtbare Zugriffsoperationen (Getter/Setter). Ergänzen Sie ebenfalls einen geeigneten Konstruktor, der es Ihnen erlaubt, die Größe bei der Instanziierung eines Fahrrades festzulegen.

Implementieren Sie in der Klasse `Fahrzeug` ein Feld „baujahr“ und entsprechende sichtbare Zugriffsoperationen. Ergänzen Sie ebenfalls geeignete Konstruktoren in den drei Klassen. Rufen Sie aus dem Konstruktor der Unterklasse den Konstruktor der Basisklasse auf (Schlüsselwort `super`). Folgender Programmcode soll möglich sein, um einen Focus mit Baujahr 1995 bzw. ein Fahrrad der Größe 28 Zoll und Baujahr 1974 anzulegen:

```
Fahrzeug golf= new Kraftfahrzeug ("Golf", 0.065, 1995);  
Fahrrad gurke= new Fahrrad(28, 1974);
```

## A.10 Hausaufgabe (toString)

[1 Punkt]

Wenn Sie diese Aufgabe bearbeiten, können Sie sie in Gestalt eines Programms zusammen für beide Aufgaben A.9 und A.10 abgeben.

Implementieren Sie die Methode „toString“ für alle drei Klassen `Fahrzeug`, `Kraftfahrzeug` und `Motorrad`. Diese Methode soll alle Eigenschaften in Form eines Strings zurückliefern, z. B. so:

- für `Fahrzeug`: „Bj. 1975“.
- für `Kraftfahrzeug`: „Bj. 2012, Tesla, 0,00 l/km“,
- für `Fahrrad`: „Bj. 1983, 28 Zoll“,

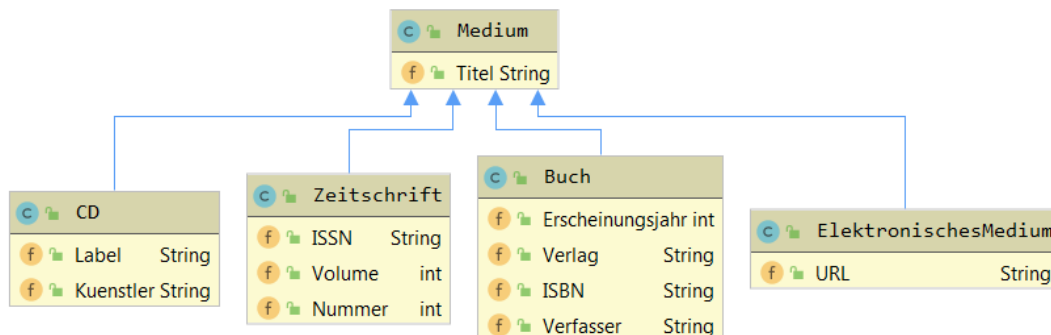
Verwenden Sie Operationen des `StringBuilder`. Informieren Sie sich über Möglichkeiten der Formatierung „Java ist auch eine Insel“ von Christian Ullenboom.

## A.11 Hausaufgabe (Klassen, abstrakte Klassen, Properties, String-Operationen, Polymorphie)

*Diese Aufgabe wird uns in den kommenden Wochen noch des Öfteren beschäftigen. Bitte bearbeiten Sie diese Aufgabe unbedingt.*

[3 Punkte]

Der Archivar einer kleinen Hochschule möchte seinen Zettelkasten digitalisieren. In seinem Zettelkasten finden sich Zettel über diverse Medien: Bücher, Zeitschriften, CDs bzw. Online-Medien. Jedes Medium hat einen Titel. Unterschiede bestehen in weiteren Daten (Künstler, Verfasser, ISBN-Nummer, Verlag, etc.). Er modelliert seinen Anwendungsbereich wie folgt:



Medium ist eine abstrakte Klasse, von der die vier unteren Klassen das Feld Titel erben.

Implementieren Sie die fünf Klassen. Schützen Sie den Zugriff auf die Felder und realisieren Sie Getter- und Setter-Methoden.

Für Bücher kennt der Bibliothekar einen Algorithmus, mit dem er die Korrektheit des ISBN prüfen kann. Implementieren Sie im Setter der ISBN die Prüfroutine, die wir am Ende dieser Aufgabe abdrucken. Im Fehlerfall geben Sie eine Meldung auf der Konsole aus. Ähnlich soll auch mit der URL für Elektronische Medien verfahren werden. Nutzen Sie hierfür die ebenfalls angegebene Prüfroutine.

Der Bibliothekar will für jeden Zettel eine Zeichenkette erstellen. Wir nennen diese Zeichenkette eine „Repräsentation“ des Mediums. Die Repräsentation des Mediums „Duden“ soll z. B. wie folgt aussehen:

Titel: Duden 01. Die deutsche Rechtschreibung  
Erscheinungsjahr: 2004  
Verlag: Bibliographisches Institut, Mannheim  
ISBN: 3-411-04013-0  
Verfasser: -

Implementieren Sie eine Methode namens „**calculateRepresentation**“, die als Rückgabewert die Repräsentation des Mediums in Gestalt einer Zeichenkette liefert. Implementieren Sie diese Methode in allen vier Subklassen. Verwenden Sie dabei den **StringBuilder**.

Schreiben Sie ein Hauptprogramm (Funktion „Main“) in einer neuen Klasse namens „Bibliothek“, das die folgenden vier Medien instanziiert und in einem Array der Länge 4 speichert.

- „Duden 01. Die deutsche Rechtschreibung“. Erscheinungsjahr: 2004. Verlag: Bibliographisches Institut, Mannheim. ISBN: 3-411-04013-0. Verfasser: -
- „1“, Label: Apple (Bea (EMI), Künstler: The Beatles
- „Der Spiegel“. ISSN 0038-7452. Volume 54. Nummer 6.
- „Hochschule Stralsund“: <http://www.hochschule-stralsund.de>

Anschließend implementieren Sie eine **for**-Schleife, die die einzelnen Repräsentationen auf der Konsole ausgibt

Erstellen Sie mit JavaDoc oder Doxygen in einem neuen Unterverzeichnis „doc“ (neben dem „bin“-Verzeichnis des Projektes) eine HTML-Version der Programmkommentare und geben diese mit ab.

Und hier nun die versprochene Prüfroutinen für die ISBN-Nummer und die URL:

```
public static boolean checkISBN10(int[] isbn) {
    int sum = 0;
    for (int i = 1; i <= isbn.length; i++) {
        sum += i * isbn[i - 1];
    }
    if (sum % 11 == 0) {
        return true;
    } else {
        return false;
    }
}

public static boolean checkISBN13(int[] isbn) {
    int sum = 0;
    for (int i = 1; i < isbn.length; i++) {
        if (i % 2 == 0) {
            sum += isbn[i - 1] * 3;
        } else {
            sum += isbn[i - 1];
        }
    }

    int lastDigit = sum % 10;

    int check = (10 - lastDigit) % 10;

    if (isbn[isbn.length - 1] == check) {
        return true;
    } else {
        return false;
    }
}

public static void main(String[] args) {
    int[] isbn10 = new int[] { 3, 8, 6, 6, 8, 0, 1, 9, 2 };
    System.out.println(checkISBN10(isbn10));
    int[] isbn13 = new int[] { 9, 7, 8, 3, 7, 6, 5, 7, 2, 7, 8, 1, 8 };
    System.out.println(checkISBN13(isbn13));
}
```

```
public static boolean checkURL(String urlString)
{
    try
    {
        URL url = new URL(urlString);
        url.toURI();
        return true;
    } catch (Exception exception)
    {
        return false;
    }
}
```