[02] Aufgabe *Algorithmen – Ablaufplan* (aus der Prüfung WS 2014/15)

Den Wert für e^x kann man über folgende Reihe nähern:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots$$

Erstellen Sie einen Ablaufplan zur Berechnung der von e^x mit einer vorgegebenen Anzahl Summanden. Das Programm soll den Wert zurückgeben und die Zahlen x und i als Parameter übergeben bekommen. Der Parameter i steht dabei für die Anzahl der Summanden.

[02] Aufgabe *Algorithmen – Wertetabelle* (aus der Prüfung WS 2014/15)

Geben Sie eine Wertetabelle für die Berechnung von e^2 mit 5 Summanden an.

[07] Aufgabe *JAVA-Klassen erstellen* (aus der Prüfung WS 2012/13)

Rechtecke lassen sich im kartesischen Koordinatensystem durch vier Attribute (rechter Rand, linker Rand, oberer Rand, unterer Rand) eindeutig beschreiben. Schreiben Sie eine JAVA-Klasse, die ein Rechteck repräsentiert und folgende Methoden enthält (s.u.). Legen Sie alle Attribute als interne Attribute an und führen für alle internen Attribute setter- und getter-Methoden aus.

- Parameterloser Konstruktor, der ein Rechteck ohne Ausdehnung im Ursprung erzeugt.
- Konstruktor mit Übergabe aller Attribute als Werte
- Berechnung des Flächeninhaltes
- Berechnung des Umfangs
- Spiegeln des Rechtecks an der X-Achse.
- Setzen aller Attribute als Werte
- Setzen der Attribute aus einem anderen Rechteck

[10] Aufgabe *Verständnis von JAVA-Quellen*

Geben Sie die Ausgaben des folgenden JAVA-Programmes auf der Konsole wieder!

```
public class Queue<ListedType> {

   private ListedType[] buffer;
   private int grow;
   private int first;
   private int length;

   public Queue(int grow) {
      this.buffer = (ListedType[]) (new ArrayList<ListedType>()).toArray();
      this.grow = grow;
```

```
    this.first = 0;
    this.length = 0;
}

public void add(ListedType value) throws Exception {
    if (first + length >= buffer.length) {
        ListedType[] tmp
            = (ListedType[]) Array.newInstance(buffer.getClass().getComponentType(),
                                               buffer.length + grow);
        arrayCopy(buffer, tmp);
        buffer = tmp;
    }

    buffer[first + length] = value;
    length++;
}

public void remove() throws Exception {
    if (length > 0) {
        buffer[first] = null;
        first++;
        length--;
    }
}

private void arrayCopy (ListedType[] source, ListedType[] destination) throws Exception {
    if (source.length <= destination.length) {
        for (int i = 0; i < source.length; i++) {
            destination[i] = source[i];
        }
    }
    else {
        throw new Exception("arrayCopy: source is longer than destination");
    }
}

public ListedType getAt(int index) throws IndexOutOfBoundsException {
    if (index >= buffer.length) {
        throw new IndexOutOfBoundsException();
    }

    return buffer[index];
}

public int getBufferLength() {
    return buffer.length;
}

public static void main(String[] args) throws Exception {
    Queue<String> q = new Queue<String>(3);

    q.add("eins");
    q.add("zwei");
```

```
        q.add("drei");

        for (int i = 0; i < q.getBufferLength(); i++)
            System.out.print(q.getAt(i) + " ");
        System.out.println();

        q.add("vier");
        q.add("fünf");

        for (int i = 0; i < q.getBufferLength(); i++)
            System.out.print(q.getAt(i) + " ");
        System.out.println();

        q.remove();
        q.remove();

        for (int i = 0; i < q.getBufferLength(); i++)
            System.out.print(q.getAt(i) + " ");
        System.out.println();

        q.remove();
        q.remove();
        q.remove();
        q.remove();

        for (int i = 0; i < q.getBufferLength(); i++)
            System.out.print(q.getAt(i) + " ");
        System.out.println();

        q.add("sech");
        q.add("sieb");

        for (int i = 0; i < q.getBufferLength(); i++)
            System.out.print(q.getAt(i) + " ");
        System.out.println();
    }

}
```

Hinweis: Falls x gleich null ist, gibt System.out.print(x + " "); die Zeichenreihe null gefolgt durch ein Leerzeichen auf der Konsole aus.

[11] Aufgabe *Erstellen eines Datenmodells* (aus der Prüfung WS 2012/13)

In einem Versandlager werden Kisten und Fässer vor dem Versand per LKW oder Zug zwischengelagert. Dort stehen die verschiedenen Versandstücke auf dem Boden und können mit Gabelstaplern aufgenommen werden. Zum besseren Wiederauffinden sind verschiedene Stellflächen eingezeichnet. Diese Stellflächen haben eine Bezeichnung wie auf einem Schachbrett (z.B. a2, b5, c7). Für den Versand werden verschiedene Versandstücke zu Versandaufträgen, die jeweils einem Kunden zugeordnet sind, zusammengefasst.

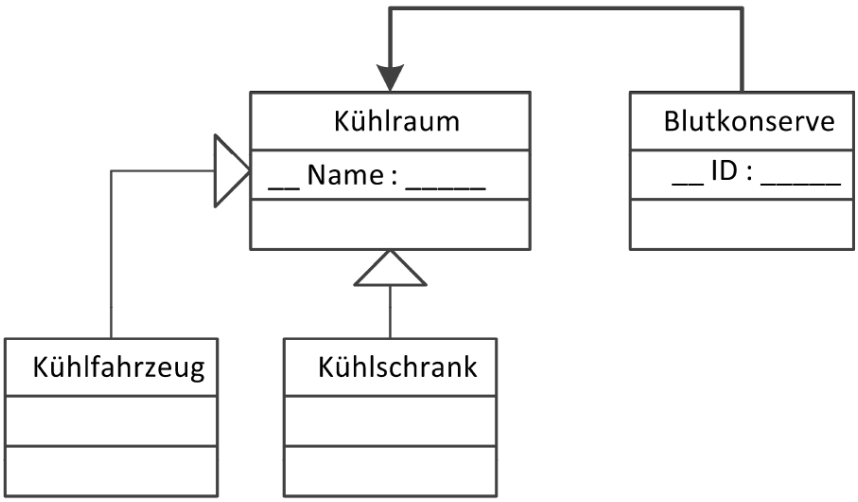
Für die Verwaltung des Lagers soll ein einfaches Datenmodell erstellt werden. Dabei sollen für Kunden jeweils Name und Kundennummer, für Kisten der Versandcode und das Gewicht und für Fässer der Versandcode, das Gewicht und der Füllstand verwaltet werden. Mit dem Modell soll es möglich sein, Versandstücke im Lager aufzufinden und einem Versandauftrag zuordnen zu können.

[11] Aufgabe *Übertragen eines Datenmodells nach JAVA* (aus der Prüfung WS 2014/15)

Zu Sicherstellung der Kühlkette beim Transport von Blutkonserven werden Kühlfahrzeuge und Kühlschränke als Lagerplätze in einem IT-System angelegt. Jede Blutkonserve liegt also immer in einem gekühlten Raum.

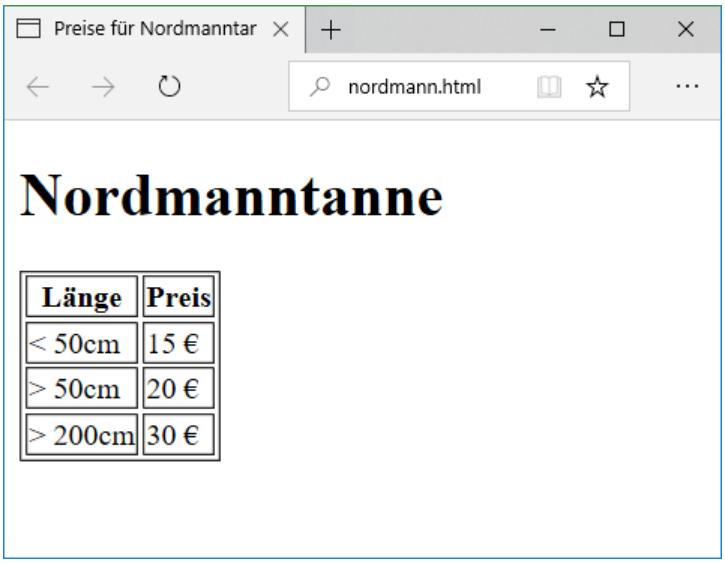
Vervollständigen Sie das folgende Datenmodell mit sinnvollen Rückgabewerten, Übergabeparametern, Kardinalitäten und Definitionen der Sichtbarkeit (public, private). Übertragen Sie das Modell anschließend in JAVA-Quellcode, legen Sie dabei alle Attribute als private an und erstellen die sinnvollen getter- und setter- Methoden, begründen Sie Ihre Entscheidung.

Beachten Sie, dass dem Konstruktor immer alle notwendigen Informationen übergeben werden müssen, damit ein gültiges Objekt erstellt wird.



[18] Aufgabe *HTML* (aus der Prüfung WS 2013/14)

Die Abbildung zeigt einen Screenshot einer HTML-Seite im Browser. Schreiben Sie die dargestellte Übersicht als einfaches HTML-Dokument.



[19] Aufgabe *Zahlensysteme* (aus der Prüfung WS 2012/13)

Vervollständigen Sie folgende Tabelle und geben Sie jeweils die Nebenrechnung an. Verwenden Sie die Restwertmethode.

Binär	Dezimal	Hexadezimal	Zur Basis 9
110010			
	77		
		CA	
			44

[20] Aufgabe *Subtraktion im Binärsystem* (aus der Prüfung WS 2014/15)

Führen Sie die Operation „59 - 12“ (neunundfünfzig minus zwölf) im binären Zahlensystem mit Hilfe des 2er-Komplementformats durch. Die Binärzahlen sollen hierbei aus 6 (binären) Ziffern bestehen. Berechnen Sie die Ausgangswerte im Binärsystem nach der Restwert-methode!

[20] Aufgabe *Rechnen im Hexadezimalsystem* (aus der Prüfung WS 2014/15)

Führen Sie die Operation „3D8 / C“ als schriftliche Division durch. Nehmen Sie dazu keine (!) Umrechnung ins Dezimalsystem vor.

[21] Aufgabe *Rechnen mit negativen Zahlen* (aus dem Video)

Berechnen Sie im Binärsystem (7 Bitstellen) mit Hilfe des 2er-Komplements –12 – 5. Führen Sie eine Addition durch!

[24] Aufgabe *Logische Schaltungen – Erstellen einer logischen Schaltung* (aus dem Video)

Zur Vereinfachung von Abstimmungsverfahren haben die fünf Weisen auf ein elektronisches Verfahren umgestellt. Bei einfachen Ja/Nein-Entscheidungen kann jeder für Ja einen Knopf drücken und für Nein drückt er nicht. Als Ergebnis leuchtet eine Lampe (Ja) oder eben nicht (Nein). Zwei der Weisen sind besonders weise. Wenn sie beide für bzw. gegen eine Entscheidung stimmen, überstimmen sie alle anderen. Sind diese beiden sich nicht einig, entscheidet die Mehrheit.

Modellieren Sie das Problem des Abstimmungsverfahrens als Schaltung, schreiben Sie den Term auf, vereinfachen diesen und geben die vereinfachte Schaltung an!

[24] Aufgabe *Logische Schaltungen – Erstellen einer Wahrheitstabelle* (aus der Prüfung WS 2014/15)

An einem Schleusentor ist ein Differenzstandmesser angebracht. Dieser liefert, wenn im Oberwasser der Wasserstand höher ist als im Schleusenbecken, das Signal '1' und bei Gleichstand das Signal '0'. Über eine Signalleitung wird dem Torcontroller angezeigt in welchen Prozessschritt sich die Schleuse befindet. Ist die Schleuse im Hochschleusvorgang, so zeigt diese Leitung das mit dem Signal '1' an. Während des Hochschleusens soll der Schieberegler im Tor solange geöffnet (Signal '1') sein, bis Gleichstand (gleicher Wasserstand) in beiden herrscht. Herrscht gleicher Wasserstand, dann soll das Tor (Signal '1') geöffnet sein andernfalls soll das Tor geschlossen sein (Signal '0').

Definieren sie die Ein- und Ausgänge einer logischen Schaltung zur Steuerung des beschriebenen Tors und geben Sie die Wahrheitstabelle an!

[25] Aufgabe *Logische Schaltungen – Erstellen eines Terms mit KV-Diagramm* (aus der Prüfung WS 2013/14)

Gegeben ist die folgende Wahrheitstabelle. Übertragen Sie die Werte in das KV-Diagramm und geben dann den Term für die Schaltung an. Vereinfachen Sie den Term soweit möglich.

E1	E2	E3	E4	A
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

	E1	E1	$\overline{E1}$	$\overline{E1}$	
E2					E4
E2					$\overline{E4}$
$\overline{E2}$					$\overline{E4}$
$\overline{E2}$					E4
	E3	$\overline{E3}$	$\overline{E3}$	E3	

[26] Aufgabe *Registermaschine* (aus der Prüfung WS 2012/13)

Erstellen Sie ein Programm für die Registermaschine, welches die kleinere der Zahlen aus Register 1 und 2 in Register 0 schreibt! Nach dem Ablauf des Programms soll in Registern 1 und 2 wieder der jeweilige Ausgangswerte stehen.

[27] Aufgabe *Registermaschine – Division* (aus dem Video)

Erstellen Sie ein Programm für die Registermaschine, welches das Ergebnis der ganzzahligen Division der Zahlen aus den Registern 2 und 3 in Register 0 schreibt! Außerdem soll in Register 1 der Rest bei ganzzahliger Division der Zahlen aus den Registern 2 und 3 stehen. Nach dem Ablauf des Programms müssen in Register 2 und 3 nicht deren Ausgangswerte stehen.

[29] Aufgabe *Kaffeeautomat* (aus der Prüfung WS 2014/15)

Ein Kaffeeautomat verfügt über einen Münzspeicher, in den über einen Einwurf 1€-Stücke und 50Cent-Stücke eingeworfen werden können. Der Automat hat drei Tasten. Einen für die Rückgabe der im Münzspeicher gespeicherten Münzen, einen für einfachen Kaffee (Preis 1€) und einen für Cappuccino (Preis 1,50€). Der Automat soll bei passend eingeworfenem Geldbetrag nach Knopfdruck das entsprechende Produkt auswerfen. Eine Rückgabe von Münzen ist nicht vorgesehen.

Entwerfen Sie einen Controller für den beschriebenen Kaffeeautomaten als endlichen Automat! Der Controller steuert für die Ausgabe des Produktes eine Brüheinheit an, deren innere Abläufe steuert der Controller nicht. Überlegen Sie dazu zunächst, welche Eingangsbefehle der Controller überhaupt benötigt und welche Ausgänge der Controller bedienen muss.

[29] Aufgabe *Notenserver* (aus der Prüfung WS 2011/12)

Server bieten Funktionen im Netzwerk an, dazu wird eine Reihe von Befehlen festgelegt, mit denen ein Client auf diese Funktionen zugreifen kann (Protokoll).

Jeder Server ist also ein Automat. Mit Hilfe eines sog. Tokenizers zerlegt der Server die Anfragen des Clients in einzelne Wörter, wobei hier ein Leerzeichen oder ein Zeilenumbruch die Wörter voneinander trennt. Bei einem Zeilenumbruch gibt der Tokenizer zunächst das Wort und dann einen NULL-String an den Server weiter.

Zur Verwaltung von Prüfungsleistungen soll ein zentraler Server die Notenverwaltung übernehmen. Der Server kann folgende Befehle verarbeiten

- **create exam <name>**

  Anlegen einer Prüfung

- **create student <name> <matrikelnummer>**

  Anlegen eines Studenten

- **setresult <nameexam> <namestudent> <grade>**

  Verbucht eine Note für einen Studenten zu einer Prüfung.

- **show**

  Zeigt eine Liste aller Prüfungen

- **show <namestudent>**

  Zeigt eine Liste aller Noten eines Studenten.

Zeichnen Sie die graphische Darstellung des Automaten!