Aufgabe 1 (12 Punkte)

Schreiben Sie eine Klasse **`public class ZinsesZins`**, die zu einem gegebenen Anfangskapital von 100,00 € die Kapitalentwicklung bei einer jährlichen nachschüssigen Verzinsung in Höhe von 2 Prozent p. a. nach n Jahren auflistet. Ausgegeben werden soll das laufende Jahr, das Kapital inklusive Zinseszins am Jahresende und die am Jahresende für das abgelaufene Jahr gutgeschriebenen Zinsen. Die ersten beiden Zeilen sollten etwa so aussehen:

1 102,00 2,00
2 104,04 2,04

Rechnen Sie ganzzahlig in Cent, die kaufmännische Rundung wollen wir vernachlässigen. Programmieren Sie die main-Methode der Testklasse **`ZinsesZins`** so aus, dass sie die Anzahl der Jahre n von der Kommandozeile übernimmt und die geforderte Tabelle ausgibt.

```
public class ZinsesZins {

  public static void main(String[] args) { // jetzt sind Sie dran!

            int n = Integer.parseInt(args[0]);
            int capital = 10000; // Cent
            int jZins;

            for (int i=1;i<=n;i++) {
                  jZins = (capital*2) / 100;
                  capital += jZins;
                  System.out.println(i+"  "+capital+"  "+jZins);

            }
      }
```

Bemerkung:
Richtiger muss die Ausgabe so aussehen

```
System.out.print(i+"  "+capital/100+",");
if (capital%100 < 10) System.out.print("0");
System.out.print(capital%100+"  "+jZins/100+",");
if (jZins%100 < 10) System.out.print("0");
System.out.println(jZins%100);
```

Aber die oben gezeigte Lösung ist auch akzeptabel.

Aufgabe 2 (8 Punkte)

Gegeben sei folgende rekursive Java-Funktion f

```
    public static int f(int n) {

          if (n < 2)
               return 1;
          else {
(1)            int h = f(n - 1);
(2)            h = h + f(n - 2);
               return h;
          }
     }
```

Welchen Wert wird n durch den Aufruf

(0) int n=f(4); zugewiesen?

Begründen Sie Ihre Antwort, indem Sie das Ergebnis jedes Aufrufs der Funktion f protokollieren. Geben Sie den Ort des Aufrufs (Markierung (0), (1) oder (2)), sowie das Argument des Aufrufes an.

Bemerkung:
Die Methode f(n) ist die n-te Fibonacci-Zahl.

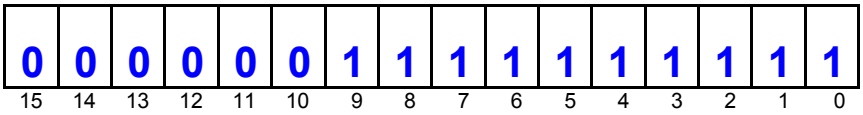
aufruf bei Markierung	Aufruf mit Parameter	Ergebnis des Aufrufes
(0)	f(4)	5
(1)	f(3)	3
(1)	f(2)	2
(1)	f(1)	1
(2)	f(0)	1
(2)	f(1)	1
(2)	f(2)	2
(1)	f(1)	1
(2)	f(0)	1

Aufgabe 3 (12 Punkte)

Wir verwenden im Folgenden die Binärdarstellung ganzer Zahlen im Zweierkomplement mit 16 Bit.

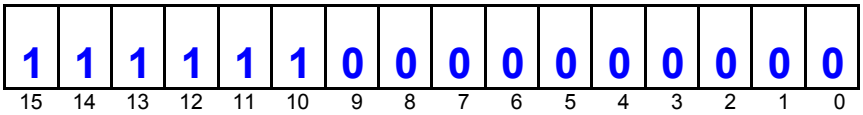
a) Geben Sie die Binärdarstellung von 1023_{10} an!
b) Geben Sie die Binärdarstellung von -1024_{10} an!
c) Welche Dezimalzahl hat die Binärdarstellung 1111111111110000 ?

a)



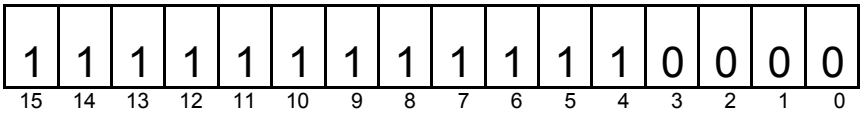
= 1023_{10}

b)



= -1024_{10}

c)



= -16

Aufgabe 4 (12 Punkte)

a) Formalisieren Sie folgende Menüregeln in Aussagenlogik, indem Sie für jeden der Sätze 1-4 eine entsprechende aussagenlogische Formel angeben. Verwenden Sie die fett gedruckten Anfangsbuchstaben als Variablennamen (z.B. B für **B**rühe)

1. Wenn eine **B**rühe gereicht wird, gibt es keinen **E**intopf
2. Nach einer **M**ehlspeise wird kein **P**udding gereicht.
3. **S**chweinebraten gibt es nur nach **B**rühe.
4. **P**udding und **B**rühe gibt es nur mit **H**uhn oder **R**inderbraten.

(1) $B \Rightarrow \neg E \quad \equiv \neg B \vee \neg E$

(2) $M \Rightarrow \neg P \quad \equiv \neg M \vee \neg P$

(3) $S \Rightarrow B \quad \equiv \neg S \vee B$

(4) $P \wedge B \Rightarrow H \vee R \quad \equiv \neg P \vee \neg B \vee H \vee R$

b) Überprüfen Sie mit Hilfe einer Wahrheitstabelle, ob $p \Rightarrow (q \Rightarrow p)$ eine Tautologie ist, d.h. für alle Belegungen der Variablen wahr ist.

p	q	$q \Rightarrow p$	$p \Rightarrow (q \Rightarrow p)$
false	*false*	***true***	***true***
false	*true*	***false***	***true***
true	*false*	***true***	***true***
true	*true*	***true***	***true***

☑ $p \Rightarrow (q \Rightarrow p)$ ist allgemeingültig

☐ $p \Rightarrow (q \Rightarrow p)$ ist nicht allgemeingültig

Aufgabe 5 (12 Punkte)

a) Implementieren Sie eine Klasse **`Datum`** mit den Attributfeldern **`tag`**, **`monat`**, **`jahr`** in Java.
b) Implementieren Sie eine Klasse **`Person`**, die **`name`**, **`vorname`**, **`strasse`**, **`postleitzahl`** und **`wohnort`** beschreiben kann.
c) Implementieren Sie unter Verwendung der Klassen **`Datum`** und **`Person`** eine Klasse **`KfzSchein`** in Java, die wichtige Eigenschaften eines zulassungspflichtigen Fahrzeuges modelliert (Beispiele jeweils in Klammern). Dazu sollen gehören: **`hersteller`** („Volkswagen“), **`typ`** („Golf 4“), **`erstzulassung`** (2.1.2003), Anzahl **`sitze`**(5), **`leergewicht`** (1.000.000 Mg), nächste **`hauptuntersuchung`** (2.1.2008), sowie der **`halter`** (Person).

```
public class Datum {
        int tag;
        int monat;
        int jahr;
}


public class Person {
        String name;
        String vorname;
        String strasse;
        String postleitzahl; // auch int postleitzahl; sinnvoll
        String wohnort;
}

public class KfzSchein {
        String hersteller;
        String typ;
        Datum erstzulassung;
        int sitze;
        int leergewicht;
        Datum hauptuntersuchung;
        Person haltr;
}
```

Aufgabe 6 **(16 Punkte)**

Diese Aufgabe umfasst 4 Multiple-Choice Cluster mit je 4 Ankreuzfragen. Für jedes Cluster gilt: wenn alle 4 Kreuze an der richtigen Stelle stehen, gibt es 4 Punkte für das Cluster. Ein falsches Kreuz gibt einen Punkt Abzug. Wer 2 richtige und 2 falsche Kreuzchen in einem Cluster macht, erhält 1+1-1-1=0 Punkte. Zum Trost: es gibt keine negativen Gesamtpunktzahlen, jedes Cluster bringt 0 bis 4 Punkte.

		Ja	Nein	Frage
a)				**Grundlagen**
	1.	☐	☑	Alle mathematischen Funktionen sind berechenbar
	2.	☑	☐	Ein Algorithmus muss mit einem *endlichen* Text beschreibbar sein
	3.	☑	☐	Alles, was mit Rekursion berechnet werden kann, kann auch ohne Rekursion (iterativ) berechnet werden
	4.	☑	☐	Durch Digitalisierung können analoge Signale in Zahlenfolgen umgesetzt werden
b)				**Java**
	1.	☑	☐	2. ist ein korrektes Java-Double Literal
	2.	☑	☐	Java verwendet Unicode.
	3.	☑	☐	Jede for-Schleife kann durch eine while-Schleife ersetzt werden
	4.	☐	☑	Die Java-Kommandos zum „Werfen“ und „Fangen“ von Exceptions heißen „drop“ und „drag“
c)				**Programmieren**
	1.	☐	☑	Auf jedes Java-Objekt darf nur eine Referenzvariable zeigen
	2.	☑	☐	Es gibt Java-Programme, die nicht terminieren
	3.	☐	☑	Die Datenstruktur „Warteschlange“ (queue) verwendet das Last-in-first-out-Prinzip
	4.	☐	☑	Es ist gut, beim Programmieren Spaghetti-Code zu schreiben
d)				**Logik**
	1.	☐	☑	$(a \wedge b) \vee (c \wedge d)$ ist in Konjunktiver Normalform
	2.	☐	☑	$\neg(a \vee b) \wedge \neg(c \vee d)$ ist in Konjunktiver Normalform
	3.	☑	☐	$a \Rightarrow a$ ist allgemeingültig
	4.	☑	☐	Für jede allgemeingültige Formel A gilt: ihr Negat $\neg A$ ist unerfüllbar