

Lernerfolgskontrolle (Java)

PPR

Nur zur Übung. Es gibt keinerlei Garantien, dass die tatsächliche Klausur den gleichen Umfang und Schwierigkeitsgrad hat.

Probeklausur

Musterlösung
Stand: 1. Februar 2014

| Aufgabe | Punkte | Erreichte Punkte |
|---------|--------|------------------|
| 1 | 8 | |
| 2 | 4 | |
| 3 | 5 | |
| 4 | 8 | |
| 5 | 4 | |
| 6 | 4 | |
| 7 | 5 | |
| 8 | 10 | |
| 9 | 4 | |
| 10 | 11 | |
| Summe | 63 | |

Aufgabe 1 (8 Punkte) Zahlensystem.

1. (3 Punkte) Berechnen Sie $15-47$ unter Verwendung der 8 Bit Zweierkomplementdarstellung. Die Rechenschritte müssen erkennbar sein.

Lösung:

$$15_{10} = 8_{10} + 4_{10} + 2_{10} + 1_{10} = 00001111_2$$

$$47_{10} = 32_{10} + 8_{10} + 4_{10} + 2_{10} + 1_{10} = 00101111_2$$

-47_{10} bilden: $47_{10} = 00101111_2$ invertieren, $\rightarrow 11010000_2$, um eins inkrementieren, $\rightarrow -47_{10} = 11010001_2$.

$$\begin{array}{r} 0|00001111 \\ + 1|11010001 \\ \hline \hline 1|11100000 \end{array}$$

1|1 \rightarrow kein Über-/Unterlauf.

$$11100000_2 = -128_{10} + 64_{10} + 32_{10} = -32_{10}$$

2. (2 Punkte) Die genetischen Informationen sind bei allen Lebewesen in den DNA-Molekülen gespeichert. Ein DNA-Molekül ist eine lineare Sequenz der vier Nukleobasen A,C,G und T. Das menschliche Genom enthält rund 2×10^9 Nukleobasen. Wie groß ist das gesamte Datenvolumen des menschlichen Genoms in Bytes?

Lösung:

$$(\log_2 4 * 2 * 10^9) / 8 = (2 * 2 * 10^9) / 8 = 500 * 10^6$$

3. (3 Punkte) Stellen Sie 30,25 in der 2 Byte binären Gleitkommadarstellung dar.
(1 Vorzeichenbit, 4 Bit Exponent: 7-Exzess-Darstellung, 11 Bit Mantisse).

Lösung:

Fixkomma: $30,25_{10} = 16_{10} + 8_{10} + 4_{10} + 2_{10} + 0,25_{10} = 11110,01_2$

Normieren: $11110,01_2 = 1,111001_2 * 2^{4_{10}} \rightarrow e = 4_{10}$, wegen 7-Exzess folgt $\rightarrow E = 11_{10} = 8_{10} + 2_{10} + 1_{10} = 1011_2$.

VZ-Bit: positiv $\rightarrow 0$, Mantisse: normierter Nachkommaanteil, aufgefüllt mit Nullen. $\rightarrow 11100100000_2$.

$\rightarrow 0\ 1011\ 11100100000$

Aufgabe 2 (4 Punkte) Logische Schaltungen.

1. (2 Punkte)

Stellen Sie die Wahrheitstabelle für eine Funktion auf, die sowohl ausgeben kann, ob eine Zahl gerade ist, als auch, ob eine Zahl ungerade ist. Sie erhält als Eingabe eine Zahl von 0 bis 3 sowie eine Eingabe, die angibt, ob die Ausgabe anzeigt, ob die Zahl gerade oder ungerade ist. Diese Eingabe erfolgt natürlich binär.

Beispiele (in Textform – eure Umsetzung dann in binär, und nur mit Zahlen von 0 bis 3!):

f(123, gerade) = falsch

f(514, gerade) = wahr

f(865, ungerade) = wahr

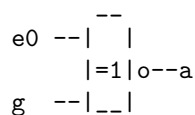
Wahrheitstabelle:

Lösung:

| <i>g</i> | <i>e</i> ₁ | <i>e</i> ₀ | <i>a</i> |
|----------|-----------------------|-----------------------|----------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

2. (2 Punkte) Zeichnen Sie eine Gatterschaltung, die obige Wahrheitstabelle umsetzt. Sämtliche bekannten Gatter dürfen verwendet werden.

Lösung:



Aufgabe 3 (5 Punkte) Rechnerarchitektur.

1. (3 Punkte)

Was versteht man unter einem *Betriebssystem*? Nennen Sie *drei Aufgaben* eines Betriebssystems.

Lösung:

- (a) Der Start und die Verwaltung von Prozessen (laufenden Programmen)
- (b) Speichermanagement
- (c) Erzeugung und Verwaltung eines logischen Dateisystems

2. (2 Punkte)

Erläutern Sie kurz die Begriffe *RAM* und *ROM*. Nennen Sie 2 Typen für RAM.

Lösung:

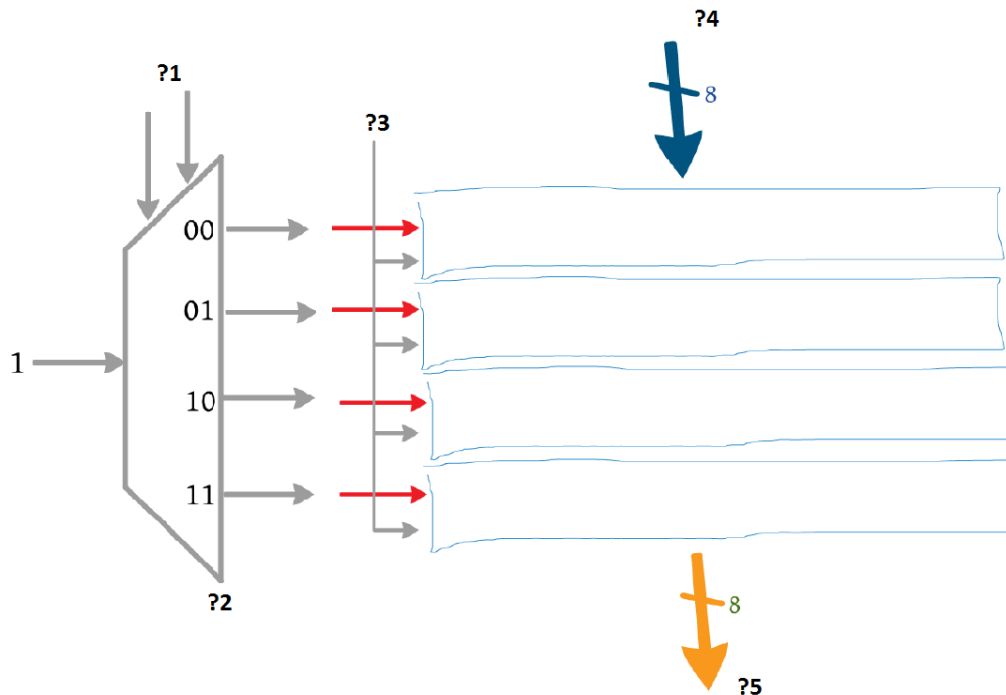
RAM ist die Abkürzung für random access memory. Der Inhalt kann jederzeit verändert werden. Es gibt statischen RAM, bestehend aus Flip-Flops, typischerweise auf Grund seiner hohen Geschwindigkeit für Cache verwendet. Dynamischer RAM, aufgebaut aus Kondensatoren, wird auf Grund seiner preiswerteren Herstellung typischerweise für den Hauptspeicher eingesetzt.

ROM bezeichnet read-only memory, dessen Inhalt fest "eingeschnitten" ist und deswegen nicht verändert werden kann.

Aufgabe 4 (8 Punkte) Rechneraufbau.

1. (4 Punkte)

Gegeben ist folgendes (aus der Vorlesung bekannte) Bild:



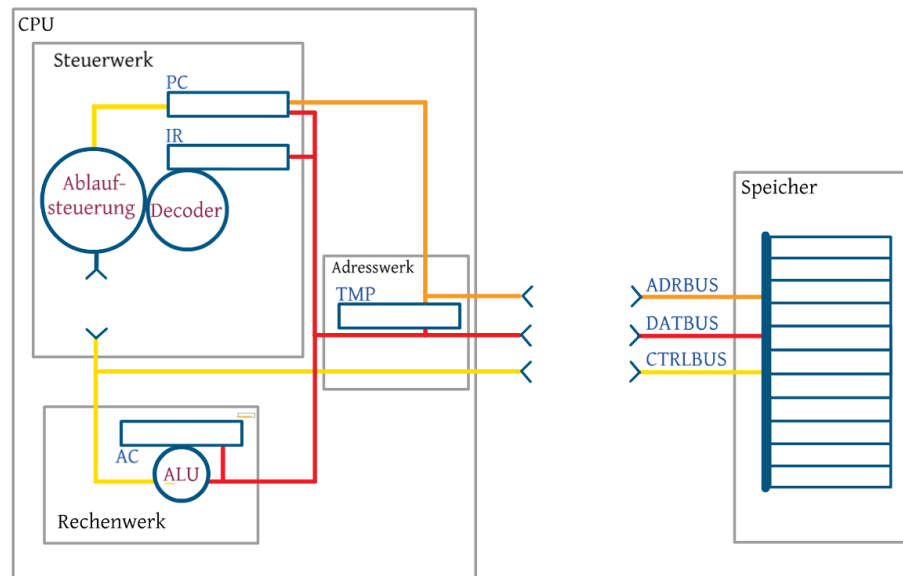
- Was stellt dies als Gesamtheit dar? (Was für eine Funktionseinheit)
- Geben Sie ?4 und ?5 vernünftige Namen und erläutern Sie die Funktion dieser Leitungen. Was bedeutet der Querstrich mit der „8“?
- Was ist die Funktion von Leitung ?3?
- Wie heißt das mit ?2 gekennzeichnete Bauteil? Wozu wird es *allgemein* (nicht nur in Bezug auf diese Schaltung) verwendet?
- Erläutern Sie die Funktion der Leitungen ?1.

Lösung:

- Dies stellt einen Tabellenspeicher dar.
- ?4: input, ?5: output. Darüber werden Daten in den Speicher geschrieben und ausgelesen. Die 8 markiert die Datenbreite dieser Leitungen (8 Parallelleitungen).
- ?3 ist die Kontrollleitung, mit der gesteuert wird, ob gelesen oder geschrieben soll.
- ?2 ist ein Demultiplexer. Es wird verwendet, um ein Eingangssignal auf verschiedene Ausgangsleitungen zu leiten, gesteuert über eine Adresse für jede Ausgangsleitung.
- Dies ist der Adresseingang für den Speicher, mit dem die jeweils zu benutzende Speicherzelle angesprochen wird.

2. (4 Punkte)

Gegeben ist wieder das folgende aus dem Tutorium bekannte vereinfachte Rechnerblockschaltbild:



a) Erklären Sie kurz und knapp die Funktion der Bauteile PC, IR, ALU, AC, ADRBUS, CTRLBUS, DATBUS.

Lösung:

PC: Program Counter. Enthält die Speicheradresse, an der sich die Programmbearbeitung gerade befindet.

IR: Instruction Register. Enthält den aktuellen Befehl.

ALU: Arithmetical Logical Unit. Führt die arithmetischen und logischen Befehle aus.

AC: Accumulator Register. Enthält (Zwischen-)Ergebnisse für Berechnungen und Operationen.

ADRBUS: Adressbus. Über ihn läuft die Adresse der aktuell benutzten Speicherzelle des Hauptspeichers.

CTRLBUS: Kontrollbus. Über ihn laufen Kontrollanweisungen zwischen den Modulen, z.B., ob im Speicher geschrieben oder gelesen werden soll.

DATBUS: Datenbus. Darüber laufen die Daten zwischen Speicher und CPU.

b) Die folgende Tabelle stellt einen Speicherausschnitt vor Beginn der darunter angegebenen Befehlsausführung dar. Der Index ₂ ist benutzt, um anzuzeigen, dass diese Zahl im Binärsystem angegeben ist.

| RAM | |
|------|-----------------------|
| 5003 | XOR |
| 5004 | 5114 |
| ... | |
| 5114 | 00001111 ₂ |

| PC | IR | AC | TMP | CTRL | CBUS | ABUS | DBUS |
|------|-----|-----------------------|------|-----------------------------------|------|------|-----------------------|
| 5003 | XOR | 01100110 ₂ | 5113 | PC++, PC→ABUS, RAM→DBUS, DBUS→IR | read | 5003 | XOR |
| 5004 | XOR | 01100110 ₂ | 5114 | PC++, PC→ABUS, RAM→DBUS, DBUS→TMP | read | 5004 | 5114 |
| 5004 | XOR | 01101001 ₂ | 5114 | TMP→ABUS, RAM→DBUS, DBUS→ALU | read | 5114 | 00001111 ₂ |

Erläutern Sie für jede Ablaufzeile die Programmausführung.

Lösung:

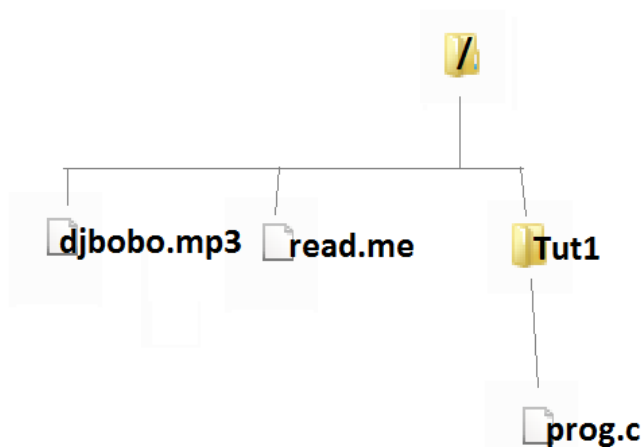
- (1) Der PC wird um eins erhöht (um den nächsten Befehl zu lesen); diese Adresse (5003) wird an den Speicher gelegt und der Inhalt der entsprechenden Zelle (XOR) über den Datenbus ins IR gelesen.
- (2) Da XOR ein Befehl mit Operand ist, wird der PC auf die nächste Adresse gesetzt; von dieser wird dann aus dem Speicher die Operandenadresse ins TMP gelesen.
- (3) Die Operandenadresse wird aus dem TMP an den Speicher gelegt, so dass der Operand über den Datenbus in die ALU gebracht werden kann, wo die Operation ausgeführt wird. Das Ergebnis landet im AC.

Aufgabe 5 (4 Punkte) Dateisystem.

1. (4 Punkte)

Das Betriebssystem speichert alle Dateien in seinem internen Dateisystem, insbesondere in der Inode-Tabelle und im Datenblock.

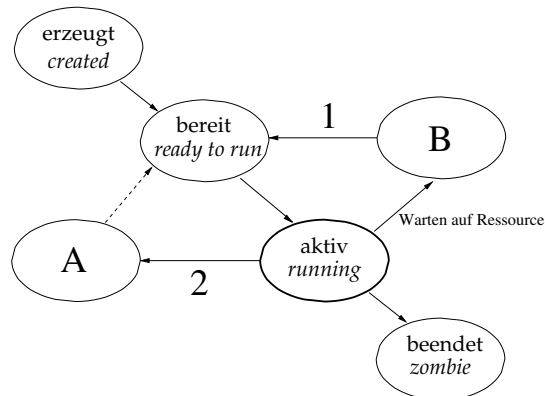
- Was wird im Datenblock abgelegt, was in der Inode-Tabelle?
- Verzeichnisse sind auch Dateien – Verzeichnisdateien. Was liegt bei einer Verzeichnisdatei im Datenblock?
- Wo findet das System den logischen Dateibaum, der dem Nutzer in Dateieexplorer-Tools angezeigt wird?
- Wieviele Inode-Ids sind in folgendem Beispiel vergeben? (Nicht sichtbare Elemente werden nicht berücksichtigt)

**Lösung:**

- In der Inode-Tabelle werden zu jeder Datei die zugehörige id, der Dateityp, die physikalische Adresse der Daten im Datenblock, und Attribute wie Erstellungsdatum, Rechte etc gespeichert. Im Datenblock liegen die eigentlichen Dateiinhalte.
- Es liegt jeweils ein Eintrag bestehend aus inode-id und Dateiname für alle beinhaltenden Dateien und Ordner sowie für den Selbstverweis und den Verweis auf das übergeordnete Verzeichnis vor.
- Dieser ist nirgendwo explizit gespeichert. Das System kann ihn aus der Inode-Tabelle und den Verzeichnisdateien inferieren.
- 5.

Aufgabe 6 (4 Punkte) UNIX.

1. (2 Punkte)



Nennen Sie die in dem Diagramm fehlenden Prozesszustände A und B (englische oder deutsche Bezeichnung), sowie die Ursachen für die Übergänge 1 und 2:

Lösung:

A: preempted

B: asleep

1: Ressource verfügbar

2: Zeitscheibe abgelaufen

2. (2 Punkte) Beantworten Sie die folgenden Fragen durch Ankreuzen von „wahr“ oder „falsch“. Jede richtige Antwort gibt 0,5 Punkte, für jede falsche Antwort werden 0,5 Punkte abgezogen. Insgesamt können aber nicht weniger als 0 Punkte erreicht werden.

Lösung:

| | <i>wahr</i> | <i>falsch</i> |
|---|-------------|---------------|
| Das Anhängen von & an einen Befehl lässt diesen im Hintergrund laufen. | (x) | () |
| Systemprioritäten haben immer Vorrang vor Benutzerprioritäten. | (x) | () |
| Mit dem Kommando nice kann jeder Benutzer die Prioritäten seiner eigenen Prozesse verbessern. | () | (x) |
| Mehrere laufende Prozesse können sich eine Prozessnummer teilen | () | (x) |

Aufgabe 7 (5 Punkte) Java: Einführung.

1. (3 Punkte)

Beantworten Sie durch Ankreuzen von „korrekt“ oder „falsch“, ob die folgenden Aussagen richtig sind. Für jedes richtig gesetzte Kreuz gibt es einen halben Punkt, für jedes falsch gesetzte Kreuz wird ein halber Punkt abgezogen. Wird kein Kreuz gesetzt, gibt es keinen Punkt und keinen Abzug.

| | <i>korrekt</i> | <i>falsch</i> |
|---|-------------------------------------|-------------------------------------|
| <code>public double int;</code> deklariert eine Kommazahl-Variable. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Bei Bezeichnern in Java wird Groß- und Kleinschreibung nicht unterschieden. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Interfaces sind Instanzen von Klassen. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Die Funktion <code>public void getValue(int a)</code> gibt einen String zurück. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Der Schleifenrumpf von <code>for (int i = 10; i >= 0; i--){...}</code> wird 10 Mal ausgeführt. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Objekt und Objektvariable ist nicht das gleiche. | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

2. (2 Punkte) Schreiben Sie für folgende Funktionalitäten jeweils einen passenden *Methodenkopf* auf.

- (a) die Methode gibt die Geschwindigkeit zurück, berechnet anhand der übergebenen Strecke (m) und Zeit (s).
- (b) die Methode gibt alle Primzahlen zwischen zwei übergebenen beliebigen Zahlen auf dem Bildschirm aus.
- (c) die Methode gibt zurück, ob eine übergebene Zahl eine Primzahl ist.

- (a) `public static float speed(float distance_m, float time_s);`
- (b) `public static void printPrimes(int a, int b);`
- (c) `public static boolean isPrime(int number);`

Aufgabe 8 (10 Punkte) Java.

1. (5 Punkte) Führen Sie eine Handsimulation durch.

- Tragen Sie die Variablenbelegungen, **nach Ausführung der Zeile** in die Tabelle ein. Achten Sie dabei auf die korrekte Darstellung des jeweiligen Typs.
- Kennzeichnen Sie aktuell nicht existierende Variablen mit `-`.
- Kennzeichnen Sie existierende Variablen, die noch nicht mit einem Wert belegt sind mit `undef`.
- Schreiben Sie bei Methodenaufrufen zusätzlich die aufrufende Zeile in Klammern hinter die momentane Zeilennummer. Bsp: Wir befinden uns in Zeile 14 in einer Funktion, die in Zeile 34 aufgerufen wurde. **Zeile: 14 (34)**
- Beachten Sie bei Methodenaufrufen, dass die erste Zeile der aufgerufenen Methode der zugehörige Methodenkopf ist.
- Nach Aufruf eines `return`-Statements ist die nächste ausgeführte Zeile der Methodenabschluss `}`.
- Die letzte Zeile einer Methode ist immer die schließende Klammer `}`.
- Die Zahl der Zeilen in der Tabelle ist abgezählt, d.h., so viele Zeilen werden im Code durchlaufen und von euch bitte beschrieben.

```

1 public class Handsimulation{
2     public static int mod(int x, int y){
3         int z = x / y;
4         if( z == 0 ){
5             return 0;
6         }
7         return x-(z*y);
8     }
9     public static void sub(int x){
10        x=x-1;
11    }
12    public static void main(String[] args){
13        int x = 3;
14        sub(x);
15        int erg = mod(x,2);
16        boolean b = (erg == x);
17    }
18 }

```

| Zeile | x(main) | erg | b | x(sub) | x(mod) | y | z |
|--------|---------|-----|-------|--------|--------|---|---|
| 12 | - | - | - | - | - | - | - |
| 13 | 3 | | | | | | |
| 9(14) | | | | 3 | | | |
| 10(14) | | | | 2 | | | |
| 11(14) | | | | - | | | |
| 14 | | | | | | | |
| 2(15) | | | | | 3 | 2 | |
| 3(15) | | | | | | | 1 |
| 4(15) | | | | | | | |
| 6(15) | | | | | | | |
| 7(15) | | | | | | | |
| 8(15) | | | | | - | - | - |
| 15 | | 1 | | | | | |
| 16 | | | False | | | | |
| 17 | - | - | - | | | | |

2. (2 Punkte)

```
public static void f(boolean a, boolean b) {
    boolean c;
    if ( !(a || b) && !(b && !a) ){
        c = true;
    }else{
        c = false;
    }
    System.out.println(c);
}
```

Geben Sie in der folgenden Tabelle an, was die Methode f bei der jeweiligen Belegung der Parameter ausgibt.

| a | b | Ausgabe von $f(a,b)$ |
|-------|-------|----------------------|
| false | false | <i>true</i> |
| false | true | <i>false</i> |
| true | false | <i>false</i> |
| true | true | <i>false</i> |

3. (3 Punkte) Schreiben Sie eine Methode `oddNumber`, die ein Array von Ganzzahlen übergeben bekommt und jede Zahl des Arrays ausgibt, falls sie ungerade ist. Beachten Sie, dass der Methode `oddNumber` beliebig lange Arrays übergeben werden könnten.

Hinweis: Gegeben sei eine Klasse `FunMath`, die eine statische Methode `isEven(int x)` beinhaltet. Die statische Methode `isEven(int x)` gibt `true` zurück, wenn es sich um eine gerade Zahl handelt, andernfalls gibt sie `false` zurück.

```
public void oddNumber(int[] zahlen){
    for(int i = 0; i<zahlen.length; i++){
        if(!FunMath.isEven(zahlen[i])){
            System.out.println(zahlen[i]);
        }
    }
}
```

Aufgabe 9 (4 Punkte) Java.

Gegeben seien folgende Java-Klassen:

```
01  class X {
02      static int s = 2;
03      void f1(){
04          System.out.println("X::f1");
05      }
06      void f2(){
07          System.out.println("X::f2");
08      }
09  }
10
11  class Y extends X {
12      static int s = 1;
13      void f1(){
14          System.out.println("Y::f1");
15      }
16      void f3(){
17          System.out.println("Y::f3");
18      }
19  }
20
21  public class Test {
22      public static void main (String[ ] args) {
23          X x = new Y();
24          Y y = new Y();
25          /***
26      }
27  }
```

Geben Sie an, was die Ausgabe der folgenden Anweisungen ist, wenn sie anstelle des Kommentars in Zeile 25 eingefügt werden. Neben der Ausgabe sind auch *Compilerfehler* bzw. *Laufzeitfehler* mögliche Antworten.

Lösung:

1. ((Y) x).f1(); \\ Y::f1
2. ((X) y).f1(); \\ Y::f1
3. System.out.println(x.s); \\ 2
4. y.f3(); \\ Y::f3
5. y.f2(); \\ X::f2
6. y.f1(); \\ Y::f1
7. Y z = new X(); z.f1(); \\ Compilerfehler
8. X z = new X(); ((Y) z).f1(); \\ Laufzeitfehler

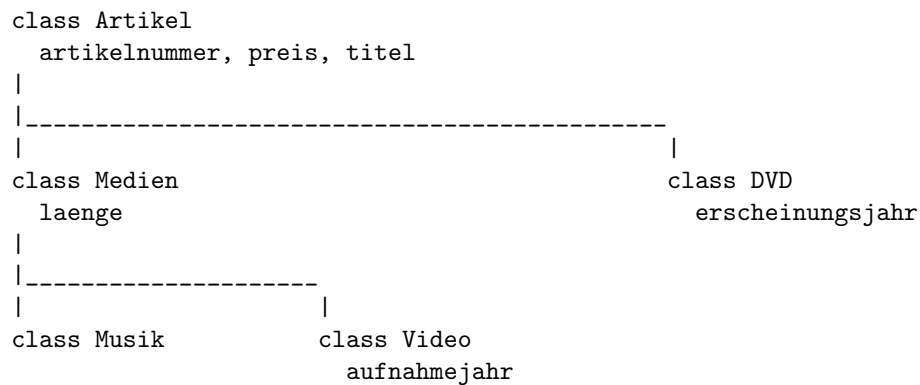
Aufgabe 10 (11 Punkte) Java.

1. (2 Punkte) Klaus verkauft Artikel bei Ebay:

- *Musik*: Beschrieben durch Artikelnummer, Preis, Titel und Länge
- *Videokassette*: Beschrieben durch Artikelnummer, Preis, Titel, Länge und Aufnahmejahr
- *DVD*: Beschrieben durch Artikelnummer, Preis, Titel und Erscheinungsjahr

Stellen Sie die Vererbungshierarchie grafisch dar. Verwenden Sie (wenn nötig) auch zusätzliche Klassen um Redundanzen zu vermeiden.

Lösung:



2. (6 Punkte) Betrachten Sie folgende Klassen und das dazugehörige Interface. Das Uboot und das Flugzeug implementieren beide das Interface Fahrzeug. Die Attribute von Uboot und Flugzeug sind nur innerhalb der Klasse sichtbar. Befüllen Sie die Lücken und implementieren Sie die unten stehende Testklasse anhand der gegebenen Kommentare.

```
public interface Fahrzeug{
    public void beschleunigen();
}
```

Lösung:

```
public class Uboot implements Fahrzeug {

    private double knoten; //Geschwindigkeit

    public Uboot(double k){
        this.knoten = k;
    }
    // Methode beschleunigen
    public void beschleunigen(){
        knoten++;
    }
}
```

```
public class Flugzeug implements Fahrzeug {

    private double kmh; //Geschwindigkeit des Flugzeugs

    // Methode beschleunigen
    public void beschleunigen(){
        kmh= ((int)Math.random()*100)*kmh;
    }
}
```

```
public class TestFahrzeug {
    public static void main (String [] args){
        // Erzeugen Sie eine ArrayListe vom Typ List
        List<Fahrzeug> fzgListe = new ArrayList<Fahrzeug>();

        // Fuegen Sie ein Uboot zur Liste hinzu
        fzgListe .add( new Uboot( 30 ) );

        // Fuegen Sie ein Flugzeug zur Liste hinzu
        fzgListe .add( new Flugzeug() );

        // Geben Sie die Laenge der Liste auf dem Bildschirm aus.
        System.out.println ( "Die Fahrzeugliste hat " + fzgListe.size() + " Elemente." );
    }
}
```

3. (3 Punkte)

Beantworten Sie durch Ankreuzen von „korrekt“ oder „falsch“, ob die folgenden Zeilen korrekten Java-Code darstellen. Jede richtige Antwort gibt 0,5 Punkte, für jede falsche Antwort werden 0,5 Punkte abgezogen. Insgesamt können aber nicht weniger als 0 Punkte erreicht werden.

| | <i>korrekt</i> | <i>falsch</i> |
|---|-------------------------------------|-------------------------------------|
| <code>boolean b = Fahrzeug instanceof Uboot;</code> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <code>Fahrzeug fz = new Fahrzeug();</code> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <code>Flugzeug ub = new Uboot();</code> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <code>Flugzeug uf = new Fahrzeug();</code> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <code>Fahrzeug ub = new Uboot();</code> | | |
| <code>ub.beschleunigen();</code> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| <code>Fahrzeug uf = new Flugzeug();</code> | | |
| <code>uf.beschleunigen();</code> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |



