

Information Technology

Prof. Dr.-Ing. Birgit Vogel-Heuser

Example examination

Duration of the exam 60 Minutes

Achievable points 60 Points

Type of exam E-Test (written) at University

Approved tools None

The exam takes place as an e-test (Platform Moodle) on a computer provided by the university. Results must be entered in the fields provided for them.

For programming tasks, a compiler is available. The compiler can check and correct your entries.

Attention:

“Prüfen”-Button submits your answers, leading to an intermediate correction and feedback regarding your task. If submitting a wrong answer, you can correct it based on the feedback to still achieve points. However, some subpoints are lost due to the wrong submission.

“Vorabprüfen”-Button runs the compiler, providing hints on possible Syntax errors. *“Vorabprüfen”* does NOT lead to loss of points, thus it is recommended prior to using *“Prüfen”-Button*.

Antwort: (Strafsystem: 10, 20, ... %)

The screenshot shows a Moodle question interface. At the top, it says 'Antwort: (Strafsystem: 10, 20, ... %)'. Below this is a text input field with the number '1' in a small box on the left. A callout box points to the input field with the text 'Input your answer here'. At the bottom of the interface are two buttons: 'Vorabprüfung' and 'Prüfen'. A callout box points to the 'Vorabprüfung' button with the text 'Syntax check by compiler without point loss **Usage Recommended!**'. Another callout box points to the 'Prüfen' button with the text 'submits your answers, leading to an intermediate correction and feedback but reduces number of max. achievable points'. A third callout box points to the 'Strafsystem' text at the top with the text '%reduction of achievable points if pressing “Prüfen” while wrong answer'.

Aufgabe / Task 1: Diskretisieren von Signalen / Signal discretization

Gegeben sind zwei Signale. Das blau markierte Signal $d(t)$ (|) ist die Diskretisierung des rot markierten Signals $f(t)$ (//) (vgl. Fig. 1.1). Beantworten Sie die folgenden Fragen.

Given are two signals. The signal $d(t)$, marked in blue (|), is the discretization of the signal $f(t)$, marked in red (//) (cf. Fig. 1.1). Answer the following questions.

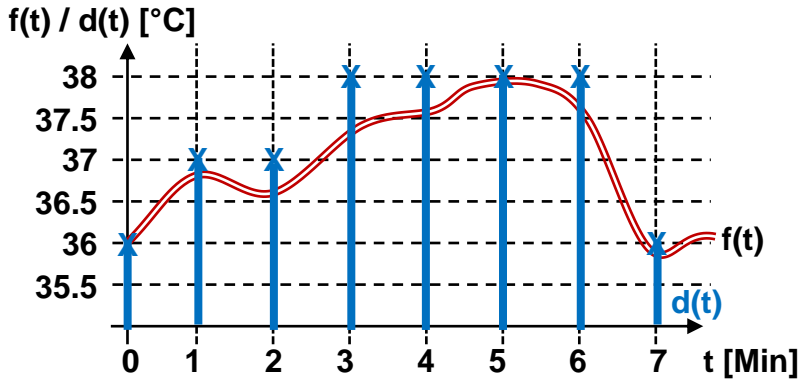


Fig.1.1: Signal $f(t)$ and its discretization $d(t)$

Das Signal $f(t)$ ist
 The signal $f(t)$ is

zeit- und wertkontinuierlich
continuous-time, continuous-value

Das Signal $d(t)$ ist
 The signal $f(t)$ is

zeit- und wertdiskret
discrete-time, discrete-value

Angenommen, das Signal $f(t)$ nimmt zum Zeitpunkt $t = 9$ den Wert 32.2 an. Welchen Wert hat das Signals $d(t)$ zu diesem Zeitpunkt?

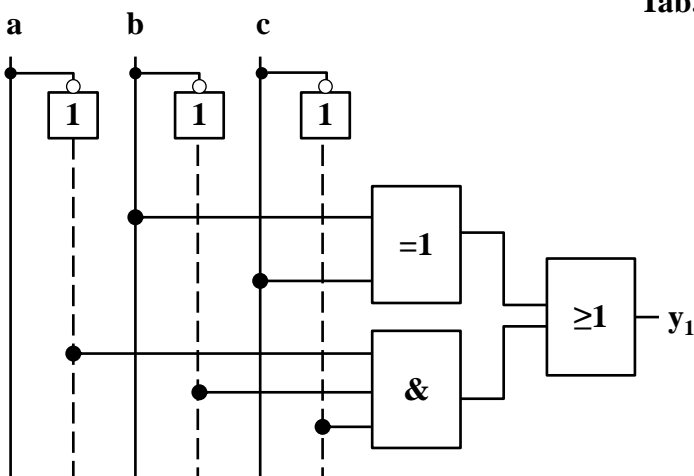
Assume that the signal $f(t)$ has the value 32.3 at time $t = 9$. What is the value of the signal $d(t)$ at this time?

33

Aufgabe / Task 2: Boolesche Algebra / Boolean algebra

Vervollständigen Sie die Wahrheitstabelle (Tab. 2.1) für die Schaltung (Fig. 2.1).

Complete the truth table (Tab. 2.1) for the circuit (Fig. 2.1).



Tab. 2.1: Wahrheitstabelle / Truth Table

a	b	c	y_1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Fig. 2.1: Schaltung / Circuit

Aufgabe / Task 3: Zahlensysteme / Number systems

Rechnen Sie die folgenden Zahlen in das jeweilige Zahlensystem (siehe Index am Ende der Klammern) um.

Convert the following numbers into the respective number system (see index at end of parentheses).

$$(0011\ 1111\ 1101)_2 = (\mathbf{3FD})_{16}$$

$$(A\ 2)_{16} = (\mathbf{162})_{10}$$

$$(10.3215)_{10} = (\mathbf{1100,0101})_2$$

Aufgabe / Task 4: Automaten / Automata

Gegeben ist das Übergangsdiagramm (Fig. 4.1) sowie die Übergangstabelle (Tab 4.1) eines Automaten. Füllen Sie passend zur Abbildung die Lücken in der Tabelle sowie die zwei blau umrandeten Lücken im Übergangsdiagramm aus. Beantworten Sie anschließend die untenstehenden Fragen. Hinweis: Achten Sie auf die Unterscheidung von "," und "/".

Given is the transition diagram and the transition table of an automaton. Fill in the gaps in the table and the two blue bordered gaps in the transition diagram to match diagram and table. Then, answer the questions below. Note: Pay attention to the distinction between "," and "/".

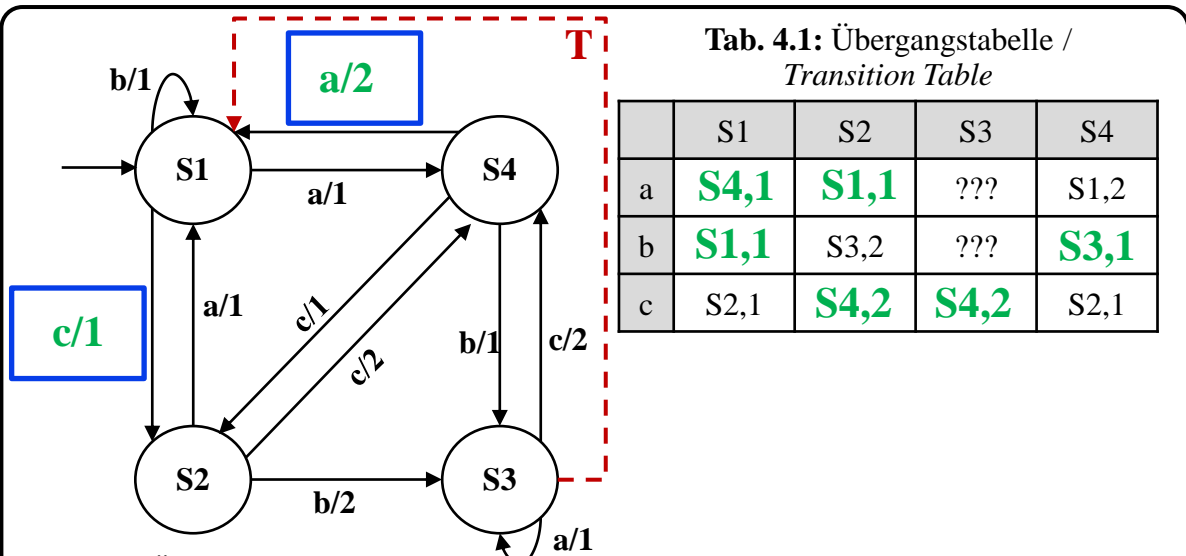


Fig. 4.1: Übergangsdiagramm / Transition Diagram

Kreuzen Sie alle Ein/Ausgang-Paare an, welche an der rot gestrichelten Transition T zulässig sind, sodass der Automat deterministisch ist.

Tick all inputs/outputs-pairs which are allowed at the red dashed transition T, so that the automaton is deterministic.

- () a/2 b/1 b/2 () c/1

Der in Fig. 4.1. abgebildete Automat ist ein **Mealy-** Automat.
 The automaton shown in Fig. 4.1. is a **Mealy-** automaton.

Aufgabe 4: Scheduling – English Version on next page

Es soll ein **präemptives Scheduling** mit festen Prioritäten verwendet werden. Bei Tasks mit **gleicher Priorität** soll ein **Round-Robin**-Verfahren mit festen Zeitschlitzen von **2 Sekunden** verwendet werden. Für jede Priorität steht eine eigene Round-Robin-Scheibe zur Verfügung. Ein Kästchen auf der x-Achse entspricht **einer Sekunde**.

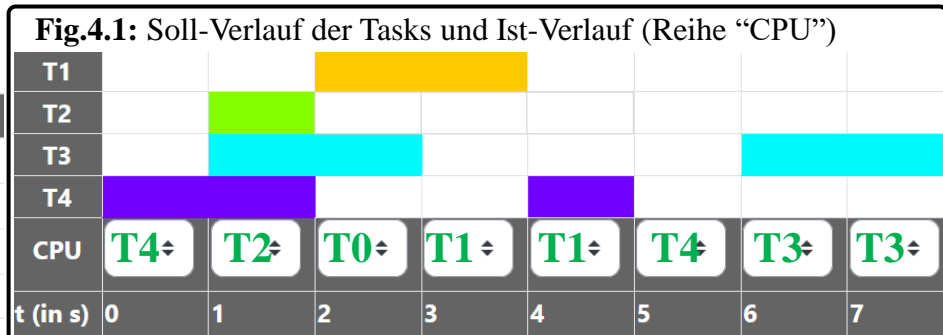
Bestimmen Sie den korrekten **Ist-Verlauf** für t im Intervall [0; 7] Sekunden zum unten gegebenen Soll-Verlauf (vgl. Fig. 4.1) der Prozesse. Tragen Sie in Fig. 4.1 in die Zeile „CPU“ die Task ein, die in dem entsprechenden Zeitabschnitt abgearbeitet wird. Tragen Sie **0** ein, falls zu einem Zeitpunkt **keine Tasks** abgearbeitet werden.

Für das Round-Robin-Verfahren sollen folgende Regeln gelten:

1. Tasks werden erst nach „Ankunft“ auf der jeweiligen Scheibe nach dem Prinzip **FIFO** (first-in-first-out) verteilt.
2. Das Scheduling beginnt erst, wenn der **erste Task anliegt** (also z.B. erst bei t=1s).
3. Wird ein Task in weniger als seiner zugeteilten 2s abgearbeitet, passiert in der **Restzeit nichts** (Tragen Sie dann **0** ein).
4. Die Scheibe mit der **höheren Priorität** kann den zugeordneten Zeitraum einer anderen Task unterbrechen. Bei erneutem Aufruf der Scheibe läuft diese **an dem unterbrochenen Zeitpunkt** weiter.
5. Die Abarbeitung der Scheiben erfolgt in Abhängigkeit von den Prioritäten: 1 (hoch) > 2 (niedrig).

Tab.4.1: Tasks mit Prioritäten

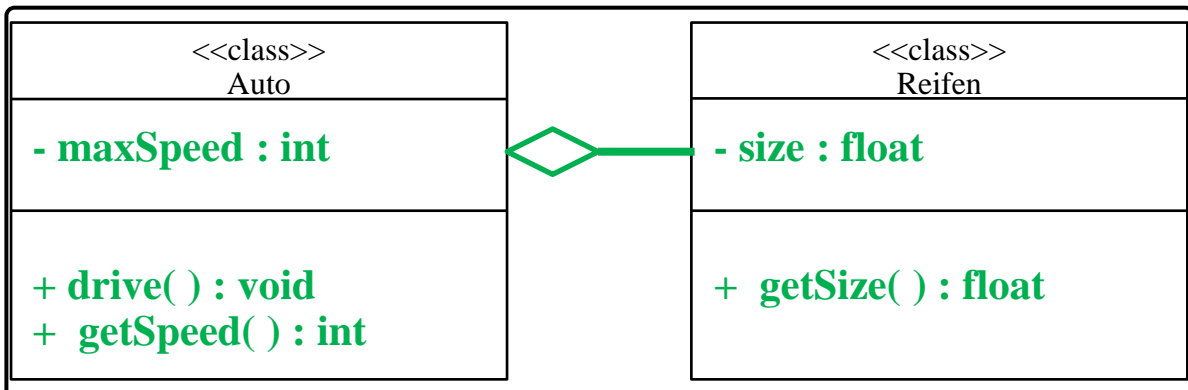
Task	Priorität
T1	1(hoch)
T2	1(hoch)
T3	2(niedrig)
T4	2(niedrig)



Aufgabe 5: UML-Klassendiagramm – English Version on next page

Gegeben sind zwei Klassen mit Namen **Auto** und **Reifen**. Ein Auto besteht aus Reifen. **Auto** hat das **Attribut maxSpeed** (Ganzzahl) und die **Methoden drive** (ohne Rückgabewert) und **getSpeed** (Wert von maxSpeed als Rückgabewert). **Reifen** haben als Attribut **Size** (Gleitkommazahl) und als Methode **getSize**, um das Attribut Size abzufragen.

Vervollständigen Sie das UML-Klassendiagramm gemäß der Beschreibung. Kardinalitäten werden vernachlässigt. Attribute sind grundsätzlich privat und Methoden öffentlich.



Task 4: Scheduling – Deutsche Version auf vorheriger Seite

Pre-emptive scheduling with fixed priorities is to be used. For tasks with the same priority, a round-robin scheduling with fixed time slots of 2 seconds is to be used. For each priority a separate round robin slice is available. One box on the x-axis corresponds to one second.

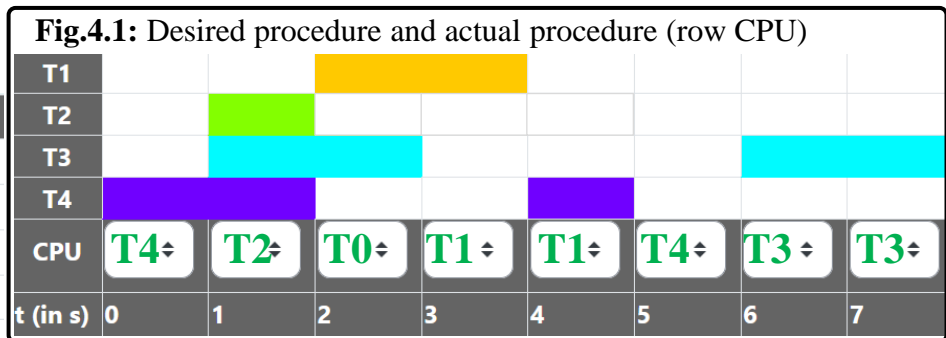
Determine the correct actual procedure for t in the interval [0; 7] seconds to the desired procedure given below (see Fig. 4.1). Enter in the line "CPU" in Fig. 4.1 the task that is processed in the corresponding time step. Enter 0 if no task is processed.

The following rules shall apply to the round robin scheduling:

- Tasks are distributed only after "arrival" on the respective slice according to the FIFO (first-in-first-out) principle.*
- Scheduling does not start until the first task is present (e.g., not until t=1s).*
- If a task is processed in less than its allocated 2s, nothing happens in the remaining time (enter 0).*
- The slice with the higher priority can interrupt the assigned time period of another task. When an interrupted slice is called again, it continues where it stopped.*
- The slices are processed depending on the priorities: Priority 1 (high) > Priority 2 (low).*

Tab.4.1: Tasks with Priority

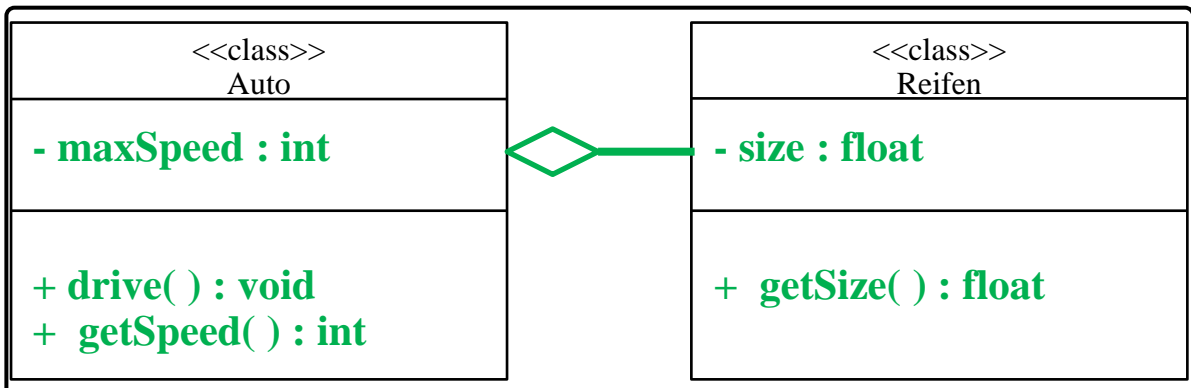
Task	Priorität
T1	1(hoch)
T2	1(hoch)
T3	2(niedrig)
T4	2(niedrig)



Task 5: UML Class Diagram – Deutsche Version auf vorheriger Seite

*Given are two classes named **Auto** and **Reifen**. **Auto** consists of **Reifen**. **Auto** has the attribute **maxSpeed** (integer) and the methods **drive** (without return value) and **getSpeed** (value of attribute maxSpeed as return value). **Reifen** has as attribute **Size** (floating point number) and as method **getSize** (returns value of attribute Size).*

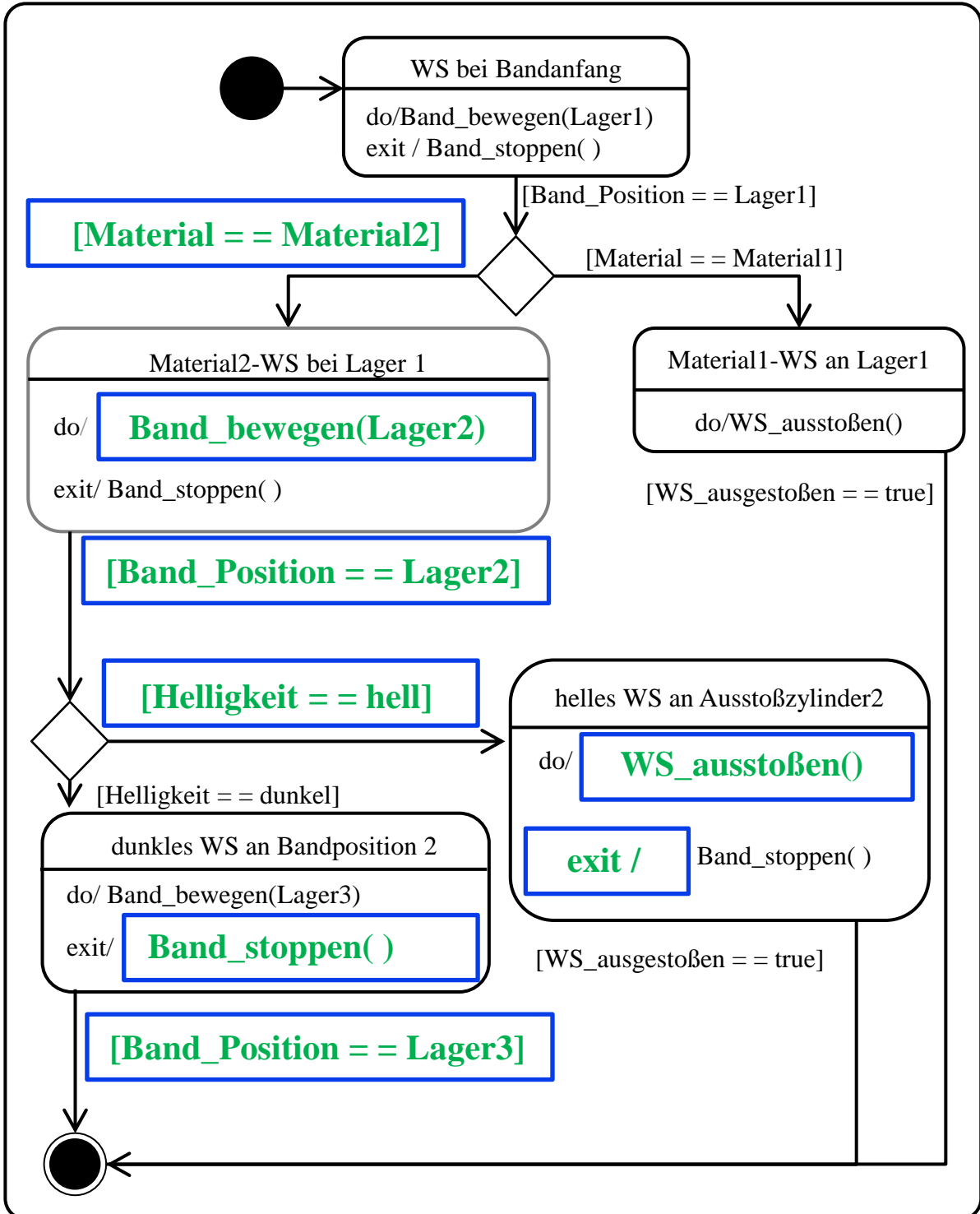
Complete the UML class diagram according to the description. Cardinalities are neglected. Attributes are always private and methods are public.



Aufgabe 5: UML-Zustandsdiagramm – English Version on next page

Vervollständigen Sie das UML-Zustandsdiagramm gemäß folgender Beschreibung:

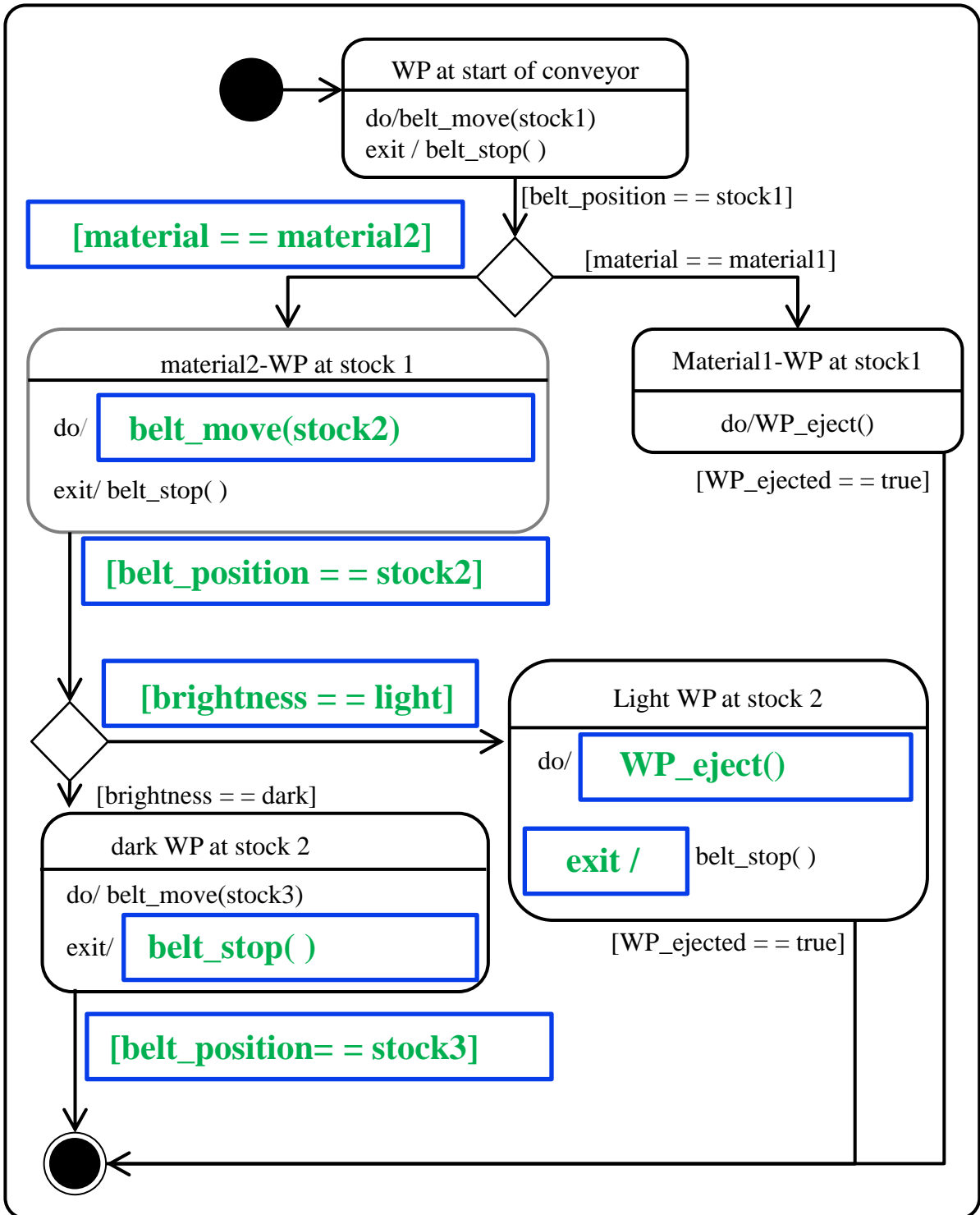
Verschiedene Werkstücke (WS) sollen je nach Material und Helligkeit in Lager sortiert werden. Der Ablauf beginnt im Zustand „WS bei Bandanfang“. Von dort transportiert das Förderband das WS zu Lager 1. Ist das WS vom Material 1, wird es ausgestoßen. Ist das WS vom Material 2, wird es zu Lager 2 weitertransportiert. An Lager 2 werden helle WS ausgestoßen und dunkle WS in Lager 3 transportiert. Das Band stoppt vor jedem Lager (Funktion **Band stoppen**). Ausstoßen erfolgt mit der Methode **WS_ausstoßen**, Transportieren erfolgt mit der Methode **Band_bewegen(Zielposition)**.



Task 5: UML State Chart – Deutsche Version auf vorheriger Seite

Complete the UML state diagram according to the following description:

Different workpieces (WP) are to be sorted into stock according to material and brightness. The process begins in the state "WP at start of conveyor". From there, the conveyor belt transports the WP to stock 1. If the WP is from material1, it is ejected. If the WP is from material 2, it is transported to stock 2. At stock 2, light WPs are ejected and dark WPs are transported to stock 3. The belt stops before each stock (function **belt_stop**). Ejecting is done with the method **WP_eject**, transporting is done with the method **belt_move(target_position)**.



Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe / Task 6: C-Programming – Schleifen / Loops

Wandeln Sie die unten gegebene for-Schleife in eine while-Schleife mit derselben Funktionalität um. Die Variablen `iGrenze` und `iErgebnis` wurden bereits deklariert. `iErgebnis` wurde mit 0 initialisiert. `iGrenze` wurde eine zuvor eingelesene Ganzzahl zugewiesen (z.B. 7).

```
for(int i = 0; i < iGrenze; i++)  
{  
    iErgebnis += 1;  
}
```

Schreiben Sie im Antwortfenster C-Programmcode mit einer while-Schleife. Dieser muss funktional identisch zur gegebenen for-Schleife sein.

Hinweise:

- Die Schleife muss zum Lösen der Aufgabe nicht unbedingt verstanden werden.
- Es ist lediglich eine while-Schleife erlaubt. Keine do-while- oder for-Schleifen!
- Es muss KEIN vollständiges C-Programm und KEINE Ein- oder Ausgabe implementiert werden.

Convert the for loop given below into a while loop with the same functionality. The variables `iGrenze` and `iErgebnis` have already been declared. `iErgebnis` has been initialized to 0. `iGrenze` has been assigned a previously read integer (e.g. 7).

```
for(int i = 0; i < iGrenze; i++)  
{  
    iErgebnis += 1;  
}
```

In the response window, write C program code with a while loop. This must be functionally identical to the given for loop.

Hints:

- *It is not necessary to understand the loop to solve the task.*
- *Only a while loop is allowed. No do-while or for loops!*
- *NO complete C program and NO input or output has to be implemented*

Antwort: (Strafsystem: 10, 20, ... %)

```
1 | int i = 0;  
  | while(i < iGrenze)  
  | {  
  |     iErgebnis += 1;  
  |     i++;  
  | }
```

Vorabprüfung

Prüfen

Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe / Task 7: C-Programming – Fallunterschiedungen / case distinctions

Schreiben Sie ein **vollständiges C-Programm**, das zwei Zahlen vom Datentyp float einliest und sie negiert, wenn sie ein positives Vorzeichen haben. Anschließend sollen sie mit zwei Nachkommastellen in der gleichen Reihenfolge, wie sie eingegeben wurden, ausgegeben werden (mit Leerzeichen getrennt).

Write a **complete C program** that reads in two numbers of the float data type and negates them if they have a positive sign. Then they are to be output with two decimal places in the same order as they were entered (separated by spaces).

Beispiel / Example:

Test	Input	Output
Test 1	-3.45 2.89	-3.45 -2.89

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 | #include <stdio.h>
   | int main(){
   |     float fZ1 = 0;
   |     float fZ2 = 0;
   |     scanf("%f", &fZ1);
   |     scanf("%f", &fZ2);
   |
   |     if(fZ1 > 0){
   |         fZ1*=-1;
   |     }
   |
   |     if(fZ2 > 0){
   |         fZ2*=-1;
   |     }
   |
   |     printf("%.2f %.2f", fZ1, fZ2);
   |     return 0;
   | }
```

Vorabprüfung

Prüfen



Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe / Task 8: C-Programming – Boolsche Algebra / Boolean Logic

Tragen Sie jeweils den Wert nach vollständiger Auswertung des Ausdrucks als **ganze Zahl** in das vorgesehene Antwortfeld hinter dem Ausdruck ein. Es wird jeweils nur das Endergebnis gewertet.

Die Variablen a,b,c sind wie unten links gezeigt deklariert und initialisiert.

*Enter the value after complete evaluation of the expression as a **whole number** in the answer field provided after the expression. Only the final result is evaluated in each case.*

The variables a,b,c are declared and initialized as shown below on the left-hand-side.

```
// Variablen:  
int a = 3;  
int b = 5;  
int c = 0;
```

<code>(a == b) && b</code>	0
<code>(c >= -- b) < a</code>	1
<code>!b c</code>	0

Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe 9: C-Programming – Call-By-Value-Funktion – English version next page

Im Folgenden soll eine **C-Funktion** entwickelt werden, die den Mindestabstand 'am' eines autonom fahrenden Autos bewertet. Ihre Funktion wird in eine bereits vorhandene Main-Funktion eingebettet, die auch die Ausgaben Ihrer Funktion erzeugt.

Programmieren Sie die C-Funktion wie folgt:

Der Rückgabewert Ihrer C-Funktion soll vom **Typ int** sein. Die Funktion muss zwingend "**mindestabstand**" heißen. Ihrer Funktion hat zwei Übergabeparameter. Dabei entspricht der erste Parameter dem **float Wert eines Sicherheitszuschlags s** (in m) und der zweite Parameter dem **Integer Wert der Geschwindigkeit v** (in km/h).

Entwickeln Sie Ihre Funktion nach folgendem Schema:

- Bei **unzulässigen Eingaben** ist die **Rückgabe** gleich **-1**
 - Unzulässige Eingaben sind der Wert Null oder negative Werte für die Übergabeparameter Sicherheitszuschlag oder Geschwindigkeit
 - Ein Wert größer gleich 256 km/h für die Geschwindigkeit ist ebenfalls unzulässig.
- Berechnen Sie den Mindestabstand in Metern mit folgender Formel: **am = v/2 + s**
- Falls der **Mindestabstand unter 10 m** liegt, ist die **Rückgabe** gleich **1**
- Falls der **Mindestabstand über 100 m** liegt, ist die **Rückgabe** gleich **2**
- In allen anderen Fällen ist die **Rückgabe** gleich **0**

Beispiel:

Eingabe	Resultat
5.53 185	Geben Sie den Sicherheitszuschlag ein: Geben Sie die momentane Geschwindigkeit ein: Mindestabstand im mittleren Bereich!

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 | int mindestabstand (float s, int v)
   | {
   |     int ire = 0;
   |     if(v >= 256 || v <= 0 || s <= 0)
   |     {
   |         ire = -1;
   |         return ire;
   |     }
   |     float am = v/2+s;
   |     if (am < 10)
   |     {
   |         ire = 1;
   |     }
   |     if (am > 100)
   |     {
   |         ire = 2;
   |     }
   |     return ire;
   | }
```

Attention: Programming Tasks are solved using a C-compiler during exam.

Task 9: C-Programming – Call-By-Value-Function – Deutsche Version vorige Seite

In the following, a **C-function** will be developed that evaluates the minimum distance 'am' of an autonomously driving car.

Your function will be embedded in an **existing main function**, which will also generate the outputs of your function. Program the C function as follows:

The return value of your C function should be of **type int**. The function must be named "**mindestabstand**". Your function has two passing parameters. The first parameter corresponds to the **float** value of a **safety margin s** (in m) and the second parameter corresponds to the **integer value of the speed v** (in km/h).

Develop your function according to the following scheme:

- For **invalid inputs**, the **return** is equal to **-1**.
 - Invalid inputs are the value zero or negative values for the transfer parameters safety margin or velocity
 - A value greater than or equal to 256 km/h for the speed is also invalid.
- Calculate the minimum distance in meters using the following equation: **am = v/2 + s**
- If the minimum distance is **less than 10m**, the return is equal to **1**
- If the minimum distance is **over 100m**, the return is equal to **2**
- In all other cases the **return** is equal to **0**

Example:

Input	Output
5.53 185	Geben Sie den Sicherheitszuschlag ein: Geben Sie die momentane Geschwindigkeit ein: Mindestabstand im mittleren Bereich!

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 |
  | int mindestabstand (float s, int v)
  | {
  |     int ire = 0;
  |     if(v >= 256 || v <= 0 || s <= 0)
  |     {
  |         ire = -1;
  |         return ire;
  |     }
  |     float am = v/2+s;
  |     if (am < 10)
  |     {
  |         ire = 1;
  |     }
  |     if (am > 100)
  |     {
  |         ire = 2;
  |     }
  |     return ire;
  | }
```

Vorabprüfung

Prüfen

Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe / Task 10: C-Programming – Zeiger/ Pointer

Realisieren Sie die folgenden Anweisungen in dieser Reihenfolge in C:

1. Deklarieren Sie zwei Zeiger mit Namen `Zeiger1` und `Zeiger2` vom Typ `Integer`
2. Setzen Sie `Zeiger1` auf den ersten Eintrag des bereits deklarierten Arrays `Array1`.
3. Setzen Sie `Zeiger2` auf den letzten Eintrag des `Array2` (Länge von `Array2` beträgt 5).
4. Schieben Sie `Zeiger1` um eine Stelle nach rechts.
5. Setzen Sie den Zeigerinhalt von `Zeiger1` auf 501.
6. Schieben Sie `Zeiger2` um 4 Stellen nach links.
7. Tauschen Sie die Zeigerinhalte der zwei Zeiger `Zeiger1` und `Zeiger2` (Hinweis: Hilfsvariable).

Implement the following statements in this order in C:

1. Declare two pointers named `Zeiger1` and `Zeiger2` of type `integer`.
2. Set `Zeiger1` to the first entry of the already declared array `Array1`.
3. Set `Zeiger2` to the last entry of `Array2` (length of `Array2` is 5).
4. Move `Zeiger1` one place to the right.
5. Set the pointer content of `Zeiger1` to 501.
6. Move `Zeiger2` 4 places to the left.
7. Swap the pointer contents of the two pointers `Zeiger1` and `Zeiger2` (hint: use an additional variable).

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 // 1.  
  int *Zeiger1;  
  int *Zeiger2;  
  
  // 2.  
  Zeiger1 = &Array1[0];  
  
  // 3.  
  Zeiger2 = &Array2[4];  
  
  // 4.  
  Zeiger1 ++;  
  
  // 5.  
  *Zeiger1 = 501;  
  
  // 6.  
  Zeiger2 = Zeiger2 - 4;  
  
  // 7.  
  int temp = *Zeiger1;  
  *Zeiger1 = *Zeiger2;  
  *Zeiger2 = temp;
```

Vorabprüfung

Prüfen

Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe / Task 11: C-Programming – Array / Array

Ihre Aufgabe ist es, ein **vollständiges C-Programm** zu schreiben, das 5 Boxen nach aufsteigender Höhe sortiert. Lesen Sie die Höhen der Boxen als separate float-Werte ein. Geben Sie die sortierte Liste mit 2 Nachkommastellen je Wert aus.

*Your task is to write a **complete C program** that sorts 5 boxes by ascending height. You get as input the heights of the boxes as float values. Output the sorted list with 2 decimal places per value.*

Beispiel / Example:	Test	Input	Output
	Test1	20.02 32.425 80.33 11.98 60.12	11.98 20.02 32.43 60.12 80.33

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 | |
#include <stdio.h>
int main(){

    float fAGewicht[5];
    int iMin = 0;
    float fTemp = 0;

    for(i = 0; i < 5; i++) {
        scanf("%f", &fAGewicht[i]);
    }

    for(i = 0; i < 4; i++) {
        iMin = i;

        for(j = i + 1; j < 5; j++) {
            if(fAGewicht[j] < fAGewicht[iMin])
            {
                iMin = j;
            }
        }

        fTemp = fAGewicht[i];
        fAGewicht[i] = fAGewicht[iMin];
        fAGewicht[iMin] = fTemp;
    }

    for(i = 0; i < 5; i++) {
        printf("%.2f\n", fAGewicht[i]);
    }
    return 0;
}
```

Attention: Programming Tasks are solved using a C-compiler during exam.

Aufgabe 12: C-Programming – File I/O – English version next page

In der folgenden Aufgabe soll eine Messreihe zur Prüfung von Abstandssensoren aufgezeichnet und in einer Textdatei gespeichert werden.

1. Erstellen Sie die Datei mit Namen "Messreihe_1_Sensoren" und Typ .txt
2. In der ersten Zeile soll "Messreihe 1 Sensoren" als string stehen.
3. In den Folgezeilen steht jeweils zunächst der maximal gemessene Abstand als float und anschließend die Anzahl an Messungen pro Minute als integer-Wert.

Der Text bzw. die Werte für die Zeilen müssen je über einen scanf-Befehl eingelesen werden. Insgesamt werden 5 Messungen durchgeführt. Folgende Grafik stellt das gewünschte Schema der Textdatei und Genauigkeit der Werte noch einmal dar. Die Zahlen sind hierbei durch ein Leerzeichen getrennt.

Beispiel:

Input	Inhalt der Datei "Messreihe_1_Sensoren.txt"
13.32	Messreihe 1 Sensoren
76	13.320 76
14.22	14.220 86
86	12.890 65
12.89	
65	

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 | int i = 0;
   | char cUeberschrift[] = "Messreihe 1 Sensoren";
   | float fAbstand = 0.0;
   | int iAnzahl = 0;
   |
   | //Erstellen der neuen Datei
   | FILE* schreiben = fopen("Messreihe_1_Sensoren.txt", "wt");
   |
   | //Überschrift schreiben
   | fputs(cUeberschrift, schreiben);
   |
   | //Schreiben in Datei
   | for(i=0;i<5;i++) {
   |     scanf("%f", &fAbstand);
   |     scanf("%d", &iAnzahl);
   | }
   |
   | //Schliessen der Schreib-Datei
   | fclose(schreiben);
```

Vorabprüfung

Prüfen

Attention: Programming Tasks are solved using a C-compiler during exam.

Task 12: C-Programming – File I/O – Deutsche Version auf vorheriger Seite

In the following task, a measurement series for testing distance sensors is to be recorded and saved in a text file.

1. Create the file with the name "Measurement series_1_Sensors" and type .txt.
2. The first line should contain the string "Measurement series 1 sensors".
3. The following lines contain the maximum measured distance as float and the number of measurements per minute as integer value.

The text or the values for the lines must each be read in via a scanf command. A total of 5 measurements are performed. The following graphic shows the desired scheme of the text file and accuracy of the values again. The numbers are separated here by a space.

Example:

Input	Content of File "Messreihe_1_Sensoren.txt"
13.32	Messreihe 1 Sensoren
76	13.320 76
14.22	14.220 86
86	12.890 65
12.89	
65	

Antwort: (Strafsystem: 10, 20, 30, 50, 100 %)

```
1 | int i = 0;
   | char cUeberschrift[] = "Messreihe 1 Sensoren";
   | float fAbstand = 0.0;
   | int iAnzahl = 0;
   |
   | //Erstellen der neuen Datei
   | FILE* schreiben = fopen("Messreihe_1_Sensoren.txt", "wt");
   |
   | //Überschrift schreiben
   | fputs(cUeberschrift, schreiben);
   |
   | //Schreiben in Datei
   | for(i=0;i<5;i++) {
   |     scanf("%f", &fAbstand);
   |     scanf("%d", &iAnzahl);
   | }
   |
   | //Schliessen der Schreib-Datei
   | fclose(schreiben);
```

Vorabprüfung

Prüfen