

Prof. Dr. Maren Bennewitz (Juniorprof.)  
M.Sc. Stefan Oßwald  
M.Sc. Eddy Ilg

Freiburg, 8. Januar 2014

# Übungsklausur – Systeme I

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

Tutor(in): \_\_\_\_\_

Umfang: 20 Seiten

Bearbeitungszeit: 70 Minuten

Erlaubte Hilfsmittel: Keine

Diese Übungsklausur wird nicht benotet und ist weder Teil der Studienleistung noch Teil der Prüfungsleistung.

Bitte prüfen Sie, ob Sie **alle Aufgabenblätter** erhalten haben und tragen Sie auf **allen** verwendeten Blättern (auch den zusätzlich ausgeteilten) Ihren **Namen** und Ihre **Matrikelnummer** ein. Blätter ohne diese Information werden nicht berücksichtigt.

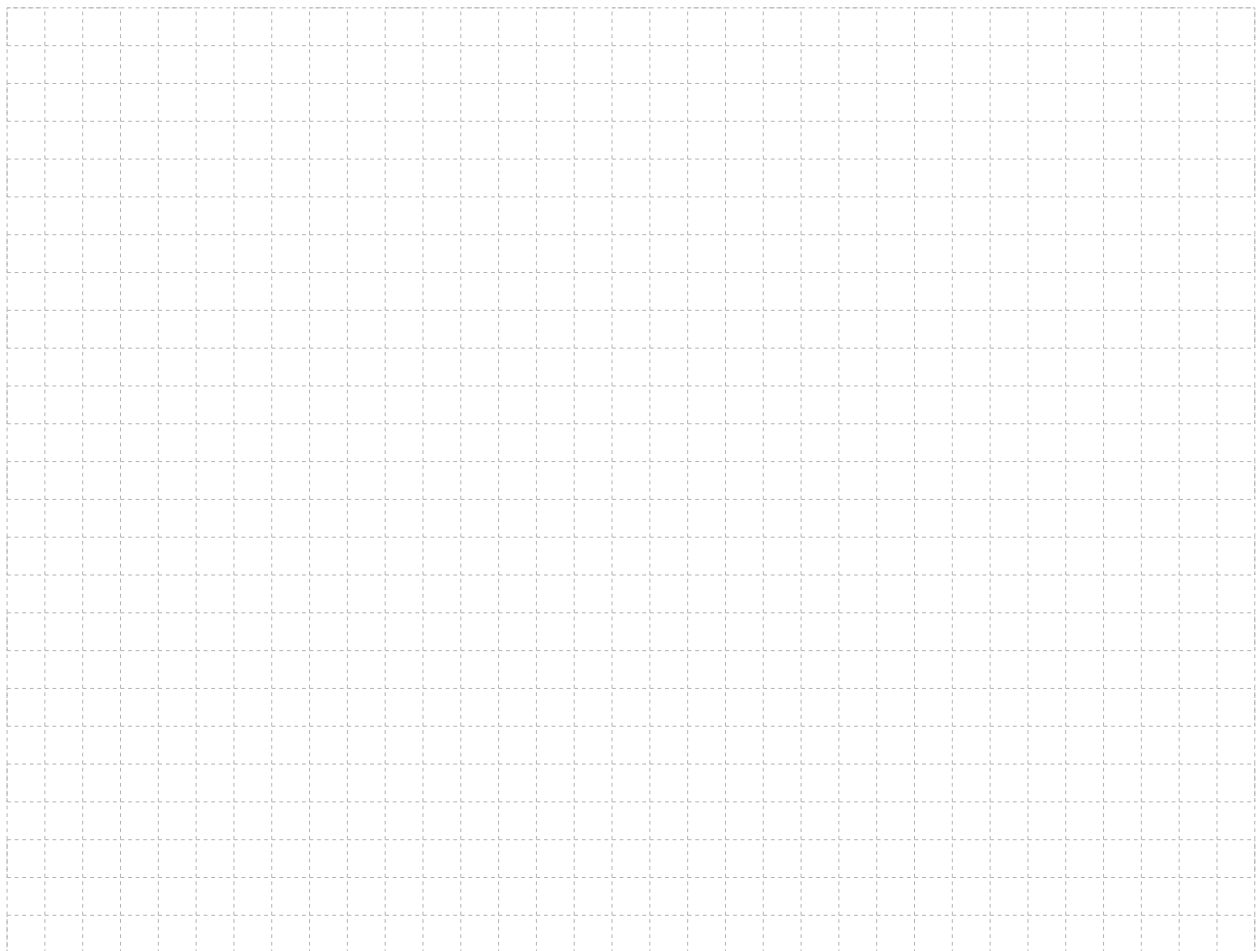
Aufgabe	Punktzahl	
	möglich	erreicht
1	7	
2	6	
3	11	
4	9	
5	6	
6	10	
7	7	
8	11	
Summe	67	

**Aufgabe 1** (2+4+1 Punkte)**Realisierung von Dateien in Dateisystemen**

In der Vorlesung wurden unter anderem „Zusammenhängende Belegung“ und „Verkettete Listen“ als Konzepte zur Realisierung von Dateien in einem Dateisystem vorgestellt.

- a) Nennen Sie die grundlegenden Unterschiede in den Arbeitsweisen der Dateiverwaltung mit „zusammenhängender Belegung“ und „verketteten Listen“.
- b) Nennen Sie insgesamt vier Eigenschaften (Vor- und Nachteile) der Dateiverwaltung mit „verketteten Listen“ gegenüber der Dateiverwaltung mit „zusammenhängender Belegung“. Begründen Sie jeden Punkt kurz.
- c) Nennen Sie ein Anwendungsgebiet, in dem Dateisysteme mit zusammenhängender Belegung sinnvoll eingesetzt werden können und begründen Sie kurz.

**Ihre Lösung zu Aufgabe 1:**

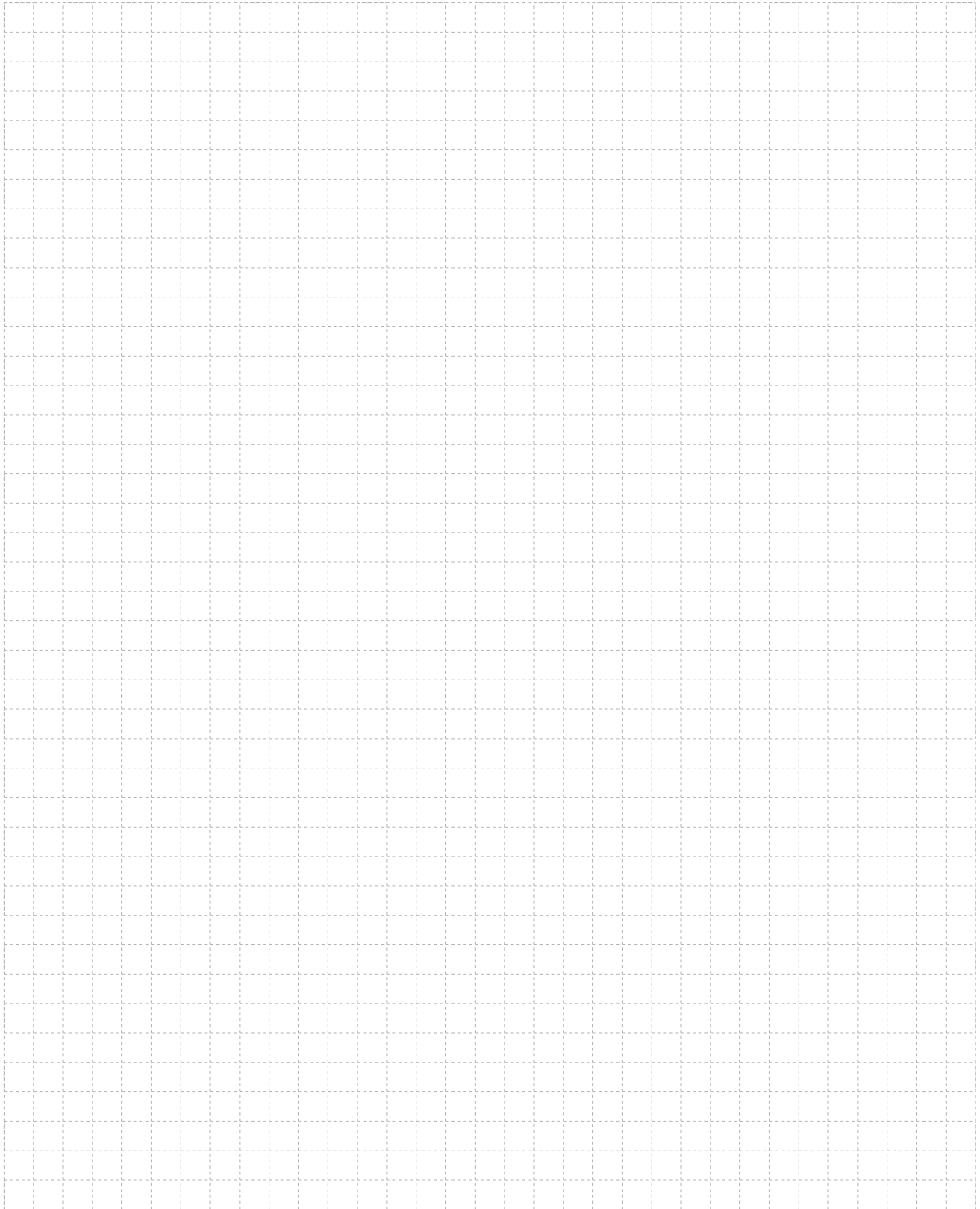
A large rectangular area filled with a grid of dashed lines, intended for the student to write their solution to the task.

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

3

Ihre Lösung zu Aufgabe 1 (Fortsetzung):



**Aufgabe 2** (3+3 Punkte)**Zugriffsrechte und Links**

Nehmen Sie an, Sie führen unter Linux den Befehl `ls -a -l` aus und erhalten dabei folgendes Ergebnis:

```
$ ls -a -l
drwxr-xr-x 4 osswald staff    4096 2013-12-16 13:29 .
drwxr-xr-x 3 root    root      0 2013-12-16 10:00 ..
drwxr-x--x 2 osswald staff    4096 2012-02-15 14:01 meine_dateien
drwxrwsrwx 2 osswald staff    4096 2012-05-03 16:17 gemeinsame_dateien
-rw-r----- 3 mueller student 38400 2014-01-07 08:02 bericht.txt
-rw-r----- 3 mueller student 38400 2014-01-07 08:02 kopie.txt
lrwxrwxrwx 1 osswald staff     11 2014-01-07 08:04 eintrag1 -> bericht.txt
```

`kopie.txt` wurde mit `ln bericht.txt kopie.txt` erstellt, ist also ein Hardlink auf die Datei `bericht.txt`.

Für die Benutzer gelten folgende Gruppenmitgliedschaften:

Benutzer	Standardgruppe	alle Gruppenmitgliedschaften
osswald	staff	staff, hrl
mueller	student	student

- a) Entscheiden Sie, ob die folgenden Aussagen richtig oder falsch sind. Für jede korrekte Antwort erhalten Sie 0,5 Punkte.

Behauptung	richtig	falsch
1. Werden die Zugriffsrechte von <code>kopie.txt</code> geändert, so ändern sich die Zugriffsrechte von <code>bericht.txt</code> automatisch mit.	<input type="checkbox"/>	<input type="checkbox"/>
2. Benutzer <code>osswald</code> bekommt eine Fehlermeldung, wenn er auf <code>eintrag1</code> lesend zugreift, z.B. mit <code>cat eintrag1</code> .	<input type="checkbox"/>	<input type="checkbox"/>
3. Wird <code>bericht.txt</code> gelöscht, ist <code>eintrag1</code> immer noch zugreifbar.	<input type="checkbox"/>	<input type="checkbox"/>
4. Wird <code>bericht.txt</code> gelöscht, ist <code>kopie.txt</code> immer noch zugreifbar.	<input type="checkbox"/>	<input type="checkbox"/>
5. Benutzer <code>mueller</code> erstellt eine neue Datei im Ordner <code>gemeinsame_dateien</code> . Dann gehört die neue Datei der Gruppe <code>student</code> .	<input type="checkbox"/>	<input type="checkbox"/>
6. Benutzer <code>mueller</code> darf die Datei <code>bericht.txt</code> löschen.	<input type="checkbox"/>	<input type="checkbox"/>

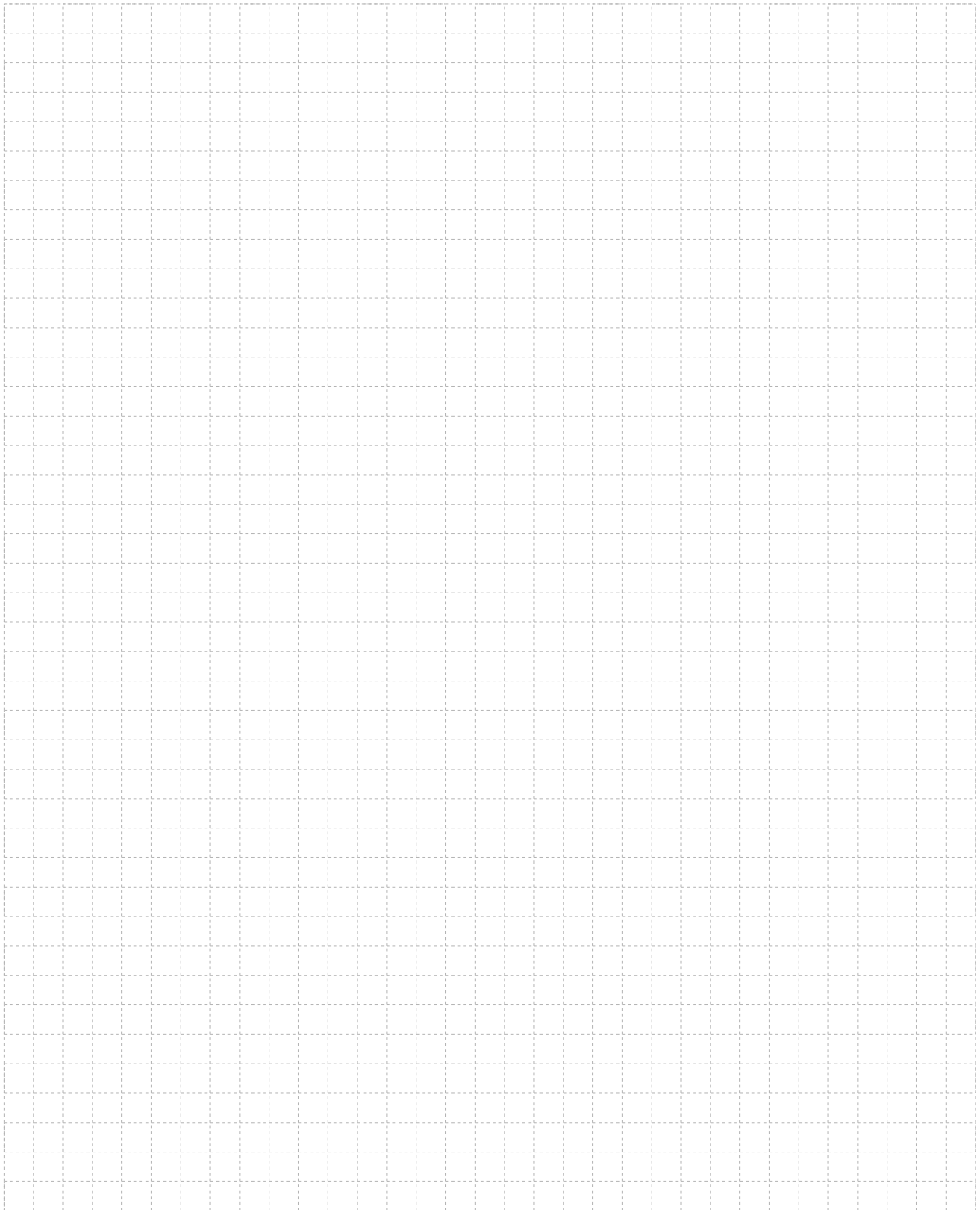
- b) Beschreiben Sie kurz, wie Hardlinks in einem Dateisystem mit I-Nodes implementiert werden. Gehen Sie insbesondere darauf ein, welche Änderungen das Betriebssystem an I-Nodes und Datenblöcken vornimmt, wenn ein Hardlink auf eine Datei angelegt bzw. gelöscht wird und wann die entsprechenden Datenblöcke wieder als „frei“ markiert werden.

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

5

Ihre Lösung zu Aufgabe 2:



**Aufgabe 3** (5+6 Punkte)**I-Node-Dateisysteme**

- a) In der Vorlesung wurden I-Nodes und ihre Struktur bei dem Betriebssystem *System V* vorgestellt:

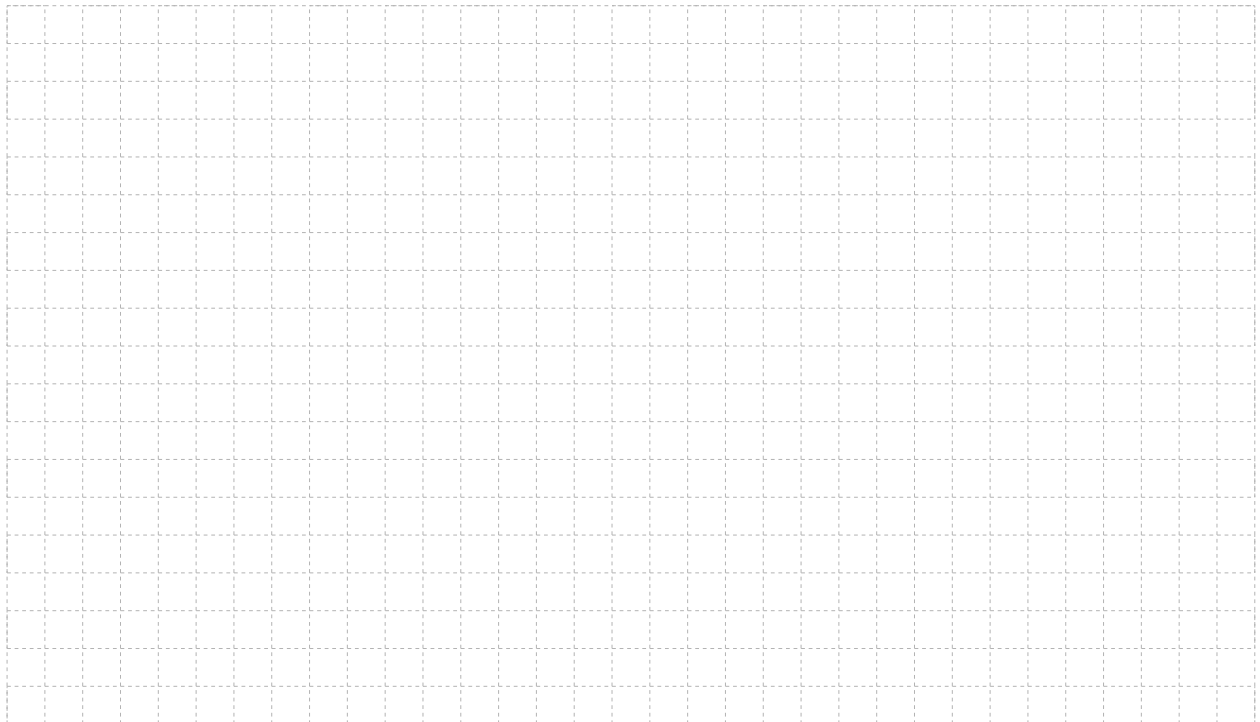
- 10 direkte Zeiger
- 1 Zeiger auf einen einfach indirekten Block
- 1 Zeiger auf einen zweifach indirekten Block
- 1 Zeiger auf einen dreifach indirekten Block

Die Blockgröße betrage 2 KB und die Zeigergröße betrage 4 Byte.

Geben Sie den Rechenweg an, um die maximal mögliche Größe einer Datei auf diesem System zu berechnen (das Endergebnis als Zahl müssen Sie nicht ausrechnen).

- b) Wie läuft ein wahlfreier Zugriff auf das Byte Nr. 20500 einer Datei bei diesem Dateisystem ab? Der entsprechende I-Node sei schon im Hauptspeicher vorhanden; die Nummerierung der Bytes fängt mit der Nummer 0 an.

Bitte geben Sie an, welche Zeiger daran beteiligt sind, an welcher Position in den Blöcken diese zu finden sind und wohin sie zeigen.

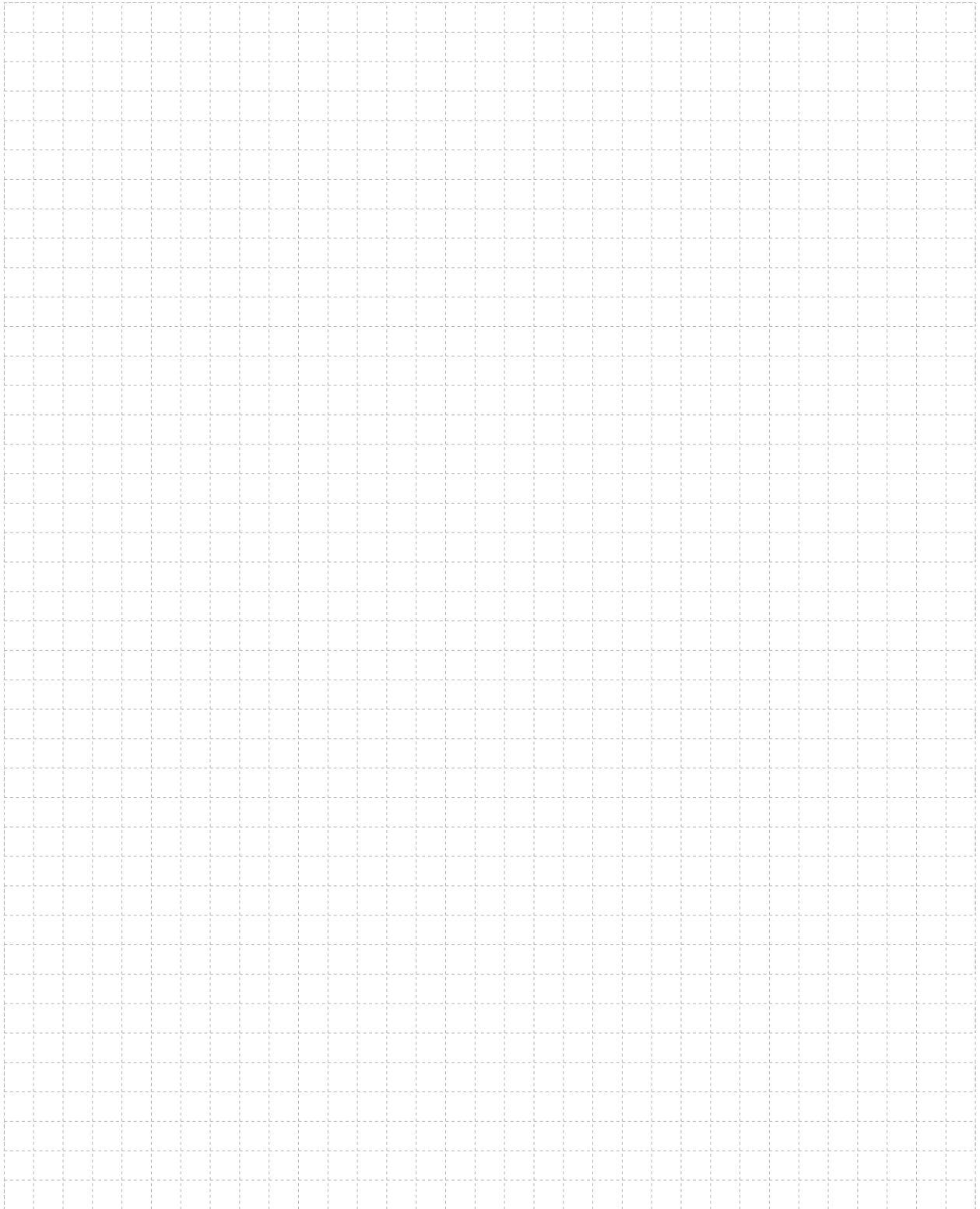
**Ihre Lösung zu Aufgabe 3:**

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

7

Ihre Lösung zu Aufgabe 3 (Fortsetzung):



**Aufgabe 4** (2+2.5+3+1.5 Punkte)**Multitasking und Prozessmodelle**

- a) Wie unterscheidet sich das präemptive vom nicht-präemptiven Prozessmodell?
- b) In der Vorlesung haben Sie fünf Prozesszustände für Prozesse im Hauptspeicher kennengelernt. Tragen Sie diese fünf Zustände in die Kreise der Abbildung 1 ein.
- c) Abbildung 1 enthält zudem Pfeile, die die Übergänge von Zuständen beschreiben. Beschriften Sie diese Pfeile entsprechend dem präemptivem Prozessmodell.
- d) Angenommen, ein rechenbereiter Prozess bekommt bei einem Prozesswechsel die CPU zugeweiht. Wie wird sichergestellt, dass die CPU das Programm des Prozesses an der richtigen Stelle fortsetzt?

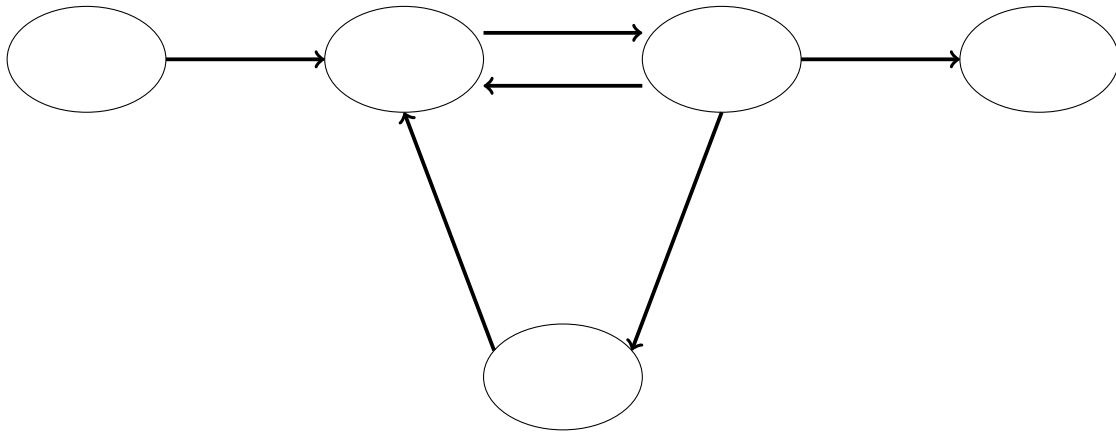


Abbildung 1: Prozessmodell mit fünf Zuständen

**Ihre Lösung zu Aufgabe 4:**

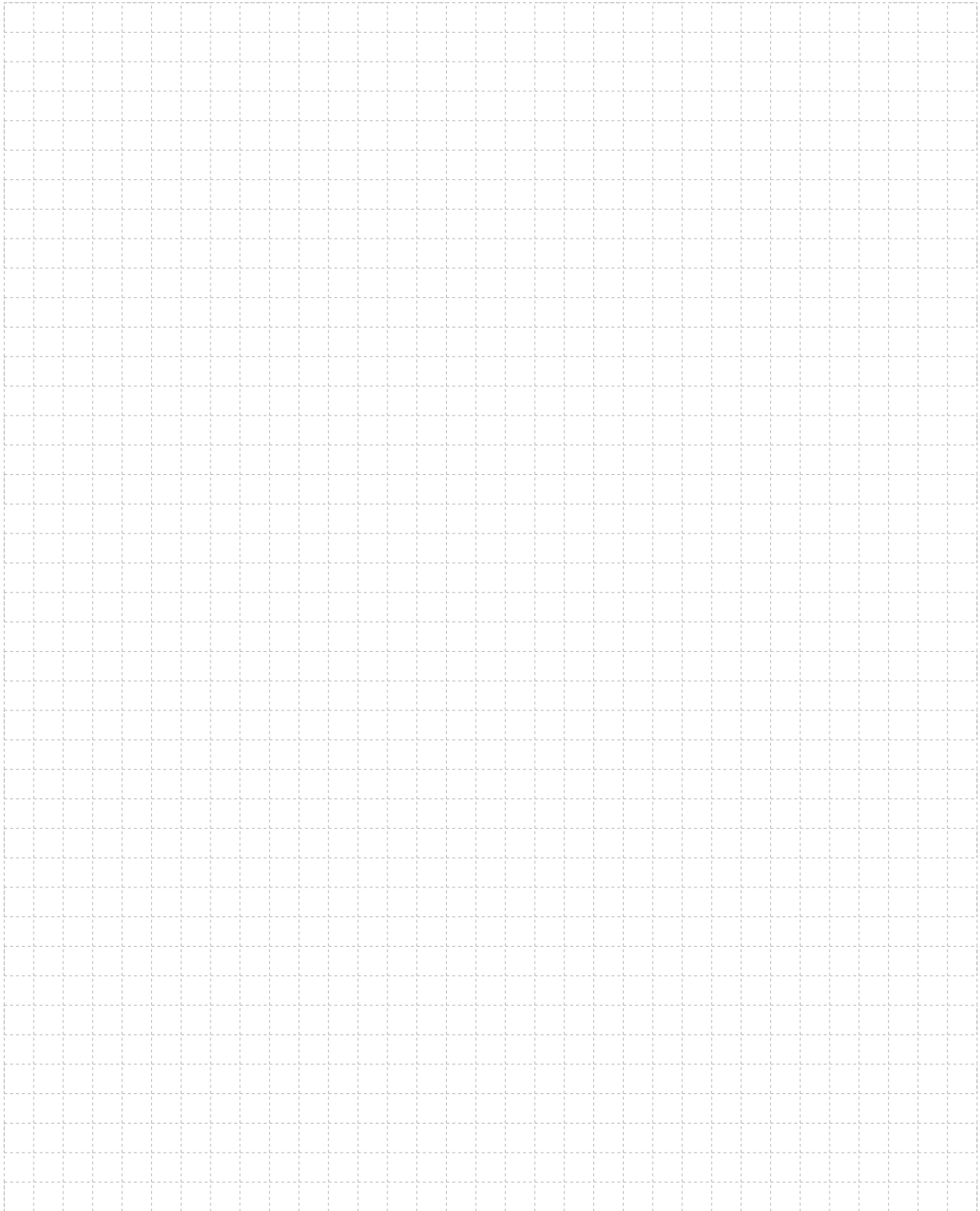


Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

9

Ihre Lösung zu Aufgabe 4 (Fortsetzung):



**Aufgabe 5** (4+2 Punkte)**Wechselseitiger Ausschluss - Petersons Algorithmus**

In der Vorlesung wurden mehrere Lösungsversuche vorgestellt, mit denen eine Softwarelösung für den wechselseitigen Ausschluss gefunden werden sollte. Hier geht es um den *Peterson-Algorithmus*:

Gemeinsame Initialisierung	
1	<code>flag[0] := false;</code>
2	<code>flag[1] := false;</code>
3	<code>turn := 0;</code>
Prozess 0	
4	<code>wiederhole</code>
5	<code>{</code>
6	<code>  flag[0] := true;</code>
7	<code>  turn := 1;</code>
8	<code>  solange (flag[1] = true und turn = 1)</code>
9	<code>    tue nichts;</code>
10	
11	<code>  Anweisung 1      }</code>
12	<code>  Anweisung 2      } <i>kritische Region</i></code>
13	<code>  ...               }</code>
14	
15	<code>  flag[0] := false;</code>
16	
17	<code>  Anweisung 3      }</code>
18	<code>  Anweisung 4      } <i>nichtkritische Region</i></code>
19	<code>  ...               }</code>
20	<code>}</code>
Prozess 1	
	<code>wiederhole</code>
	<code>{</code>
	<code>  flag[1] := true;</code>
	<code>  turn := 0;</code>
	<code>  solange (flag[0] = true und turn = 0)</code>
	<code>    tue nichts;</code>
	<code>  Anweisung 5      }</code>
	<code>  Anweisung 6      } <i>kritische Region</i></code>
	<code>  ...               }</code>
	<code>  flag[1] := false;</code>
	<code>  Anweisung 7      }</code>
	<code>  Anweisung 8      } <i>nichtkritische Region</i></code>
	<code>  ...               }</code>
	<code>}</code>

Die Anweisungen in der kritischen bzw. nichtkritischen Region ändern nichts an den Variablen `flag[0]`, `flag[1]` und `turn`.

Im Folgenden soll per Widerspruchsbeweis gezeigt werden, dass diese Lösung den wechselseitigen Ausschluss auf die kritische Region garantiert, d.h. dass die beiden Prozesse niemals gleichzeitig in der kritischen Region sein können.

Nehmen Sie dazu an, dass die Prozesse 0 und 1 zu einem Zeitpunkt  $t$  beide in der kritischen Region seien. Die Zeitpunkte, zu denen die Prozesse 0 und 1 die `solange`-Schleife zuletzt verlassen haben, seien  $t_0$  und  $t_1$ . Ohne Beschränkung der Allgemeinheit sei  $t_0 > t_1$ .

Der Beweis beruht auf einer Fallunterscheidung darüber, aus welchem Grund Prozess 0 die `solange`-Schleife zum Zeitpunkt  $t_0$  verlassen konnte. Den Fall `turn`  $\neq$  1 brauchen Sie an dieser Stelle nicht zu betrachten.

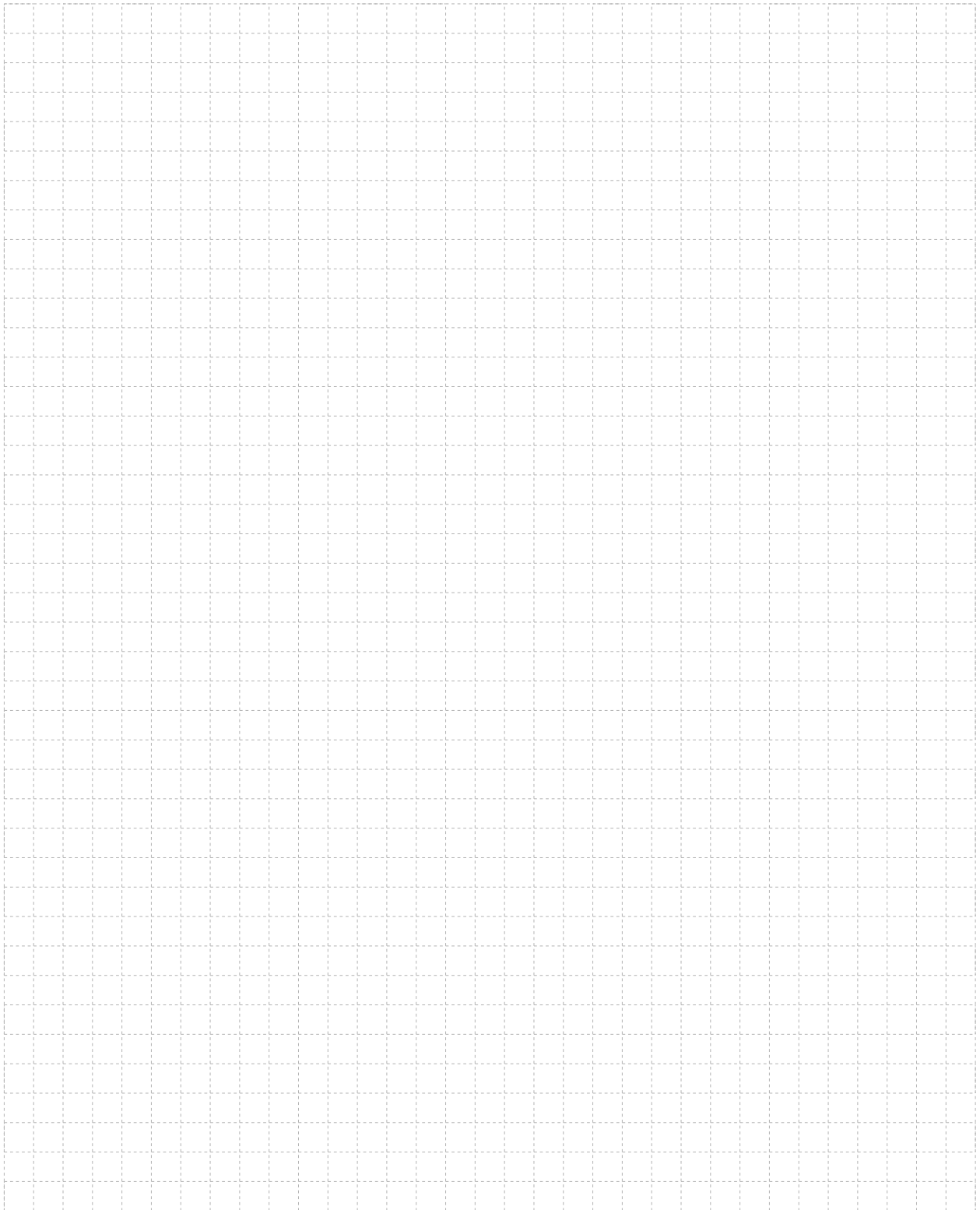
- Betrachten Sie den Fall, dass `flag[1] = false` zum Zeitpunkt  $t_0$ . Zeigen Sie, dass diese Annahme zum Widerspruch führt.
- Welchen Nachteil hat Petersons Lösungsversuch? Wie kann man diese Art von Nachteil umgehen (allgemein, nicht auf Petersons Algorithmus bezogen)?

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

11

Ihre Lösung zu Aufgabe 5:



**Aufgabe 6** (10 Punkte)**Produzenten/Konsumenten-Problem**

In der Vorlesung haben Sie das *Produzenten/Konsumenten-Problem* kennengelernt. Sie sehen hier eine Variante der Lösung aus der Vorlesung. Es wurden lediglich bei der Prozedur **producer** die Reihenfolge der Befehle `down(empty);` und `down(mutex);` vertauscht:

```
Semaphore mutex; countmutex := 1;
Semaphore empty; countempty := MAX_BUFFER;
Semaphore full; countfull := 0;
```

Prozedur producer

```
{
  wiederhole
  {
    item := produce_item();

    down(mutex);      /* Reihenfolge */
    down(empty);      /* vertauscht */

    insert_item(item);

    up(mutex);
    up(full);
  }
}
```

Prozedur consumer

```
{
  wiederhole
  {
    down(full);
    down(mutex);

    item := remove_item();

    up(mutex);
    up(empty);

    consume_item(item);
  }
}
```

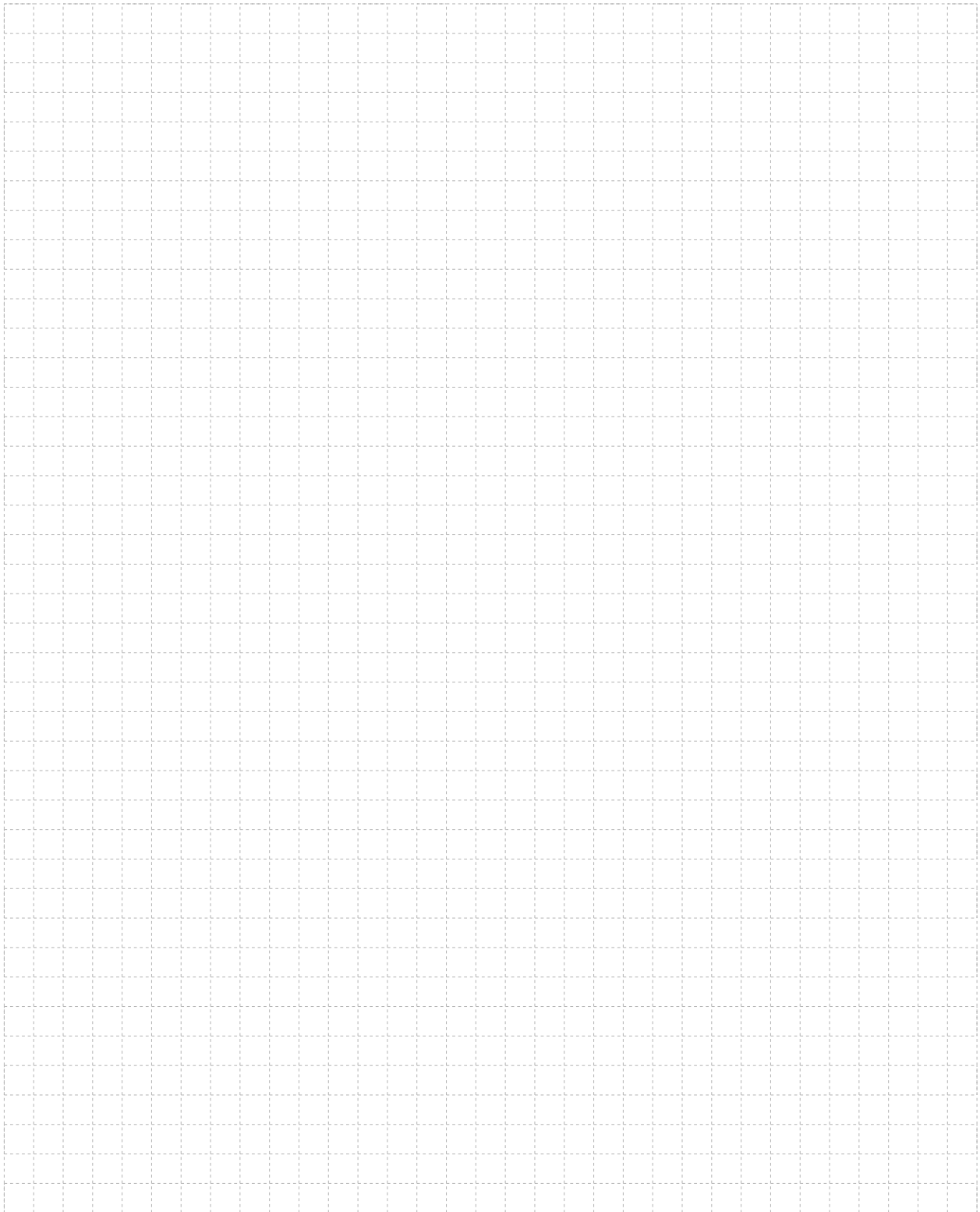
Funktioniert diese Variante der ursprünglichen korrekten Lösung fehlerfrei? Beweisen Sie entweder, dass Deadlocks bei diesem Algorithmus garantiert ausgeschlossen sind, oder geben Sie eine Ausführungsreihenfolge an, die zu einem Deadlock führt.

Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

13

Ihre Lösung zu Aufgabe 6:



**Aufgabe 7** (2+3+2 Punkte)**Deadlocks**

Zwei Prozesse wollen auf vier Ressourcen A, B, C und D zugreifen. Folgende Tabelle zeigt, in welcher Reihenfolge die Prozesse Ressourcen anfragen bzw. freigeben; Befehle, die zwischen den Anforderungen und Freigaben stehen, vernachlässigen wir hier.

Prozess 1	Prozess 2
1: Anforderung B	1: Anforderung A
2: Anforderung A	2: Anforderung D
3: Anforderung C	3: Freigabe A
4: Freigabe B	4: Anforderung B
5: Anforderung D	5: Anforderung C
6: Freigabe C	6: Freigabe B
7: Freigabe D	7: Freigabe C
8: Freigabe A	8: Freigabe D

- Zeichnen Sie in das Diagramm in Abbildung 2 auf Seite 15 die Bereiche ein, in der beide Prozesse auf eine Ressource zugreifen würden. Die horizontale Achse repräsentiert den Programmfortschritt von Prozess 1 und die vertikale Achse repräsentiert den Programmfortschritt von Prozess 2. Die mit einer Nummer versehenen horizontalen und vertikalen Linien sind die Zeitpunkte in der Programmausführung, an der die entsprechend nummerierte Zeile des Prozesses ausgeführt wird.
- In diesem Szenario kann es zu einem Deadlock kommen. Zeichnen Sie den Bereich ein, in dem ein Deadlock unvermeidlich ist und markieren Sie die Stelle, an dem der Deadlock eintritt. Begründen Sie jeweils kurz Ihre Antworten.
- Geben Sie eine Ausführungsreihenfolge an, die deadlockfrei ist, und eine, die zu einem Deadlock führt.

**Ihre Lösung zu Aufgabe 7:**

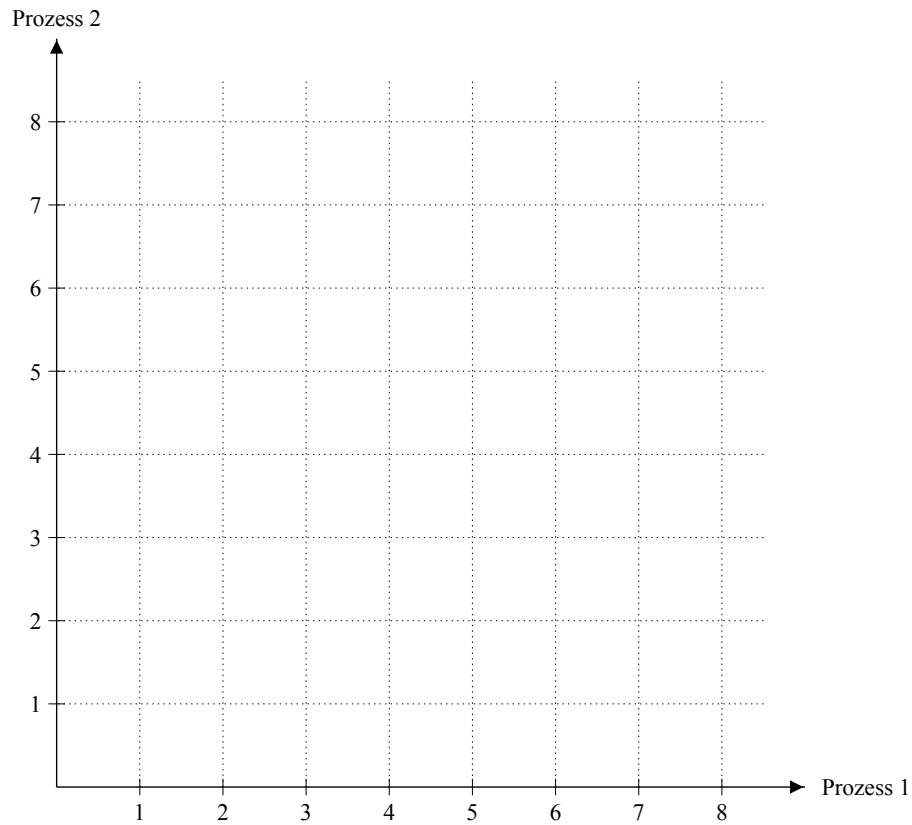
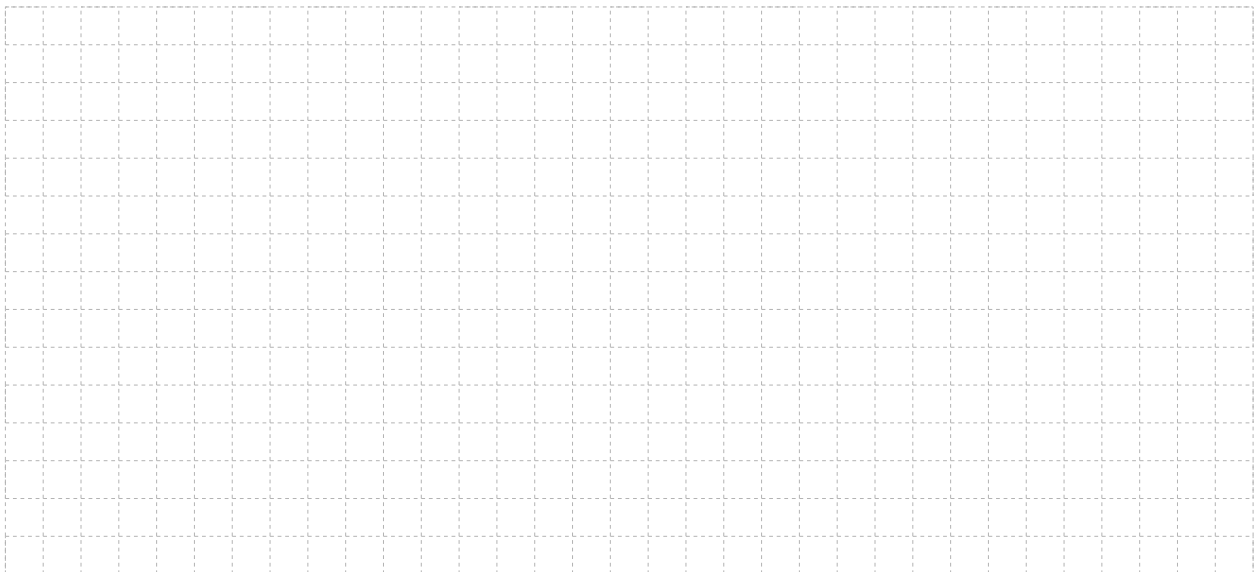


Abbildung 2: Zeichnen Sie hier die Bereiche ein.

**Ihre Lösung zu Aufgabe 7 (Fortsetzung):**



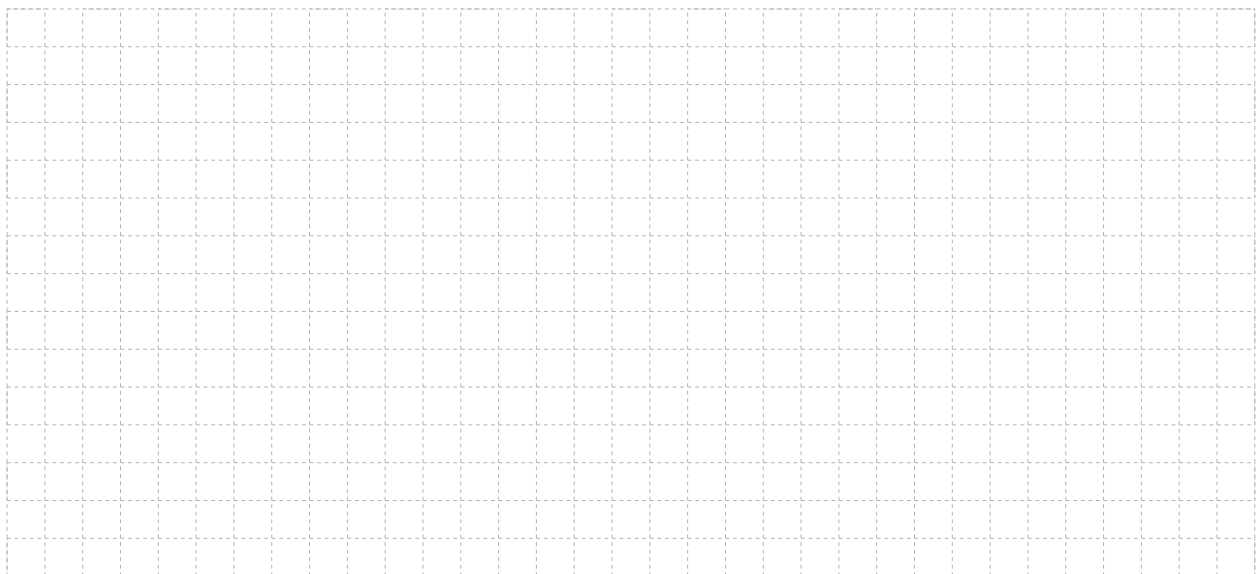
**Aufgabe 8** (4+3+2+2 Punkte)**Deadlocks und Bankier-Algorithmus**

- a) Nennen Sie die vier Voraussetzungen, die erfüllt sein müssen, damit ein Deadlock auftreten kann und erklären Sie jeden Punkt kurz.
- b) Beim Bankieralgorithmus wurde der Begriff des „sicheren Zustands“ verwendet. Wie lautet die Definition eines sicheren Zustandes?
- c) Führt jeder unsichere Zustand unweigerlich in einen Deadlock? Begründen Sie Ihre Antwort.
- d) Drei Prozesse  $p_1$ ,  $p_2$  und  $p_3$  greifen auf Ressourcen einer einzigen Ressourcenklasse zu. Insgesamt stehen  $V = 7$  Ressourcen zur Verfügung. Für die maximale Anzahl  $M_i$  von Ressourcen, auf die die Prozesse zugreifen werden, und für die Anzahl von Ressourcen  $E_i$ , die die Prozesse schon erhalten haben, gilt:

	$M_i$	$E_i$
$p_1$	6	2
$p_2$	3	2
$p_3$	6	2

Ist dieser Zustand ein „sicherer Zustand“?

- ☐ Ja, der Zustand ist sicher.
- ☐ Nein, der Zustand ist unsicher.

**Ihre Lösung zu Aufgabe 8:**

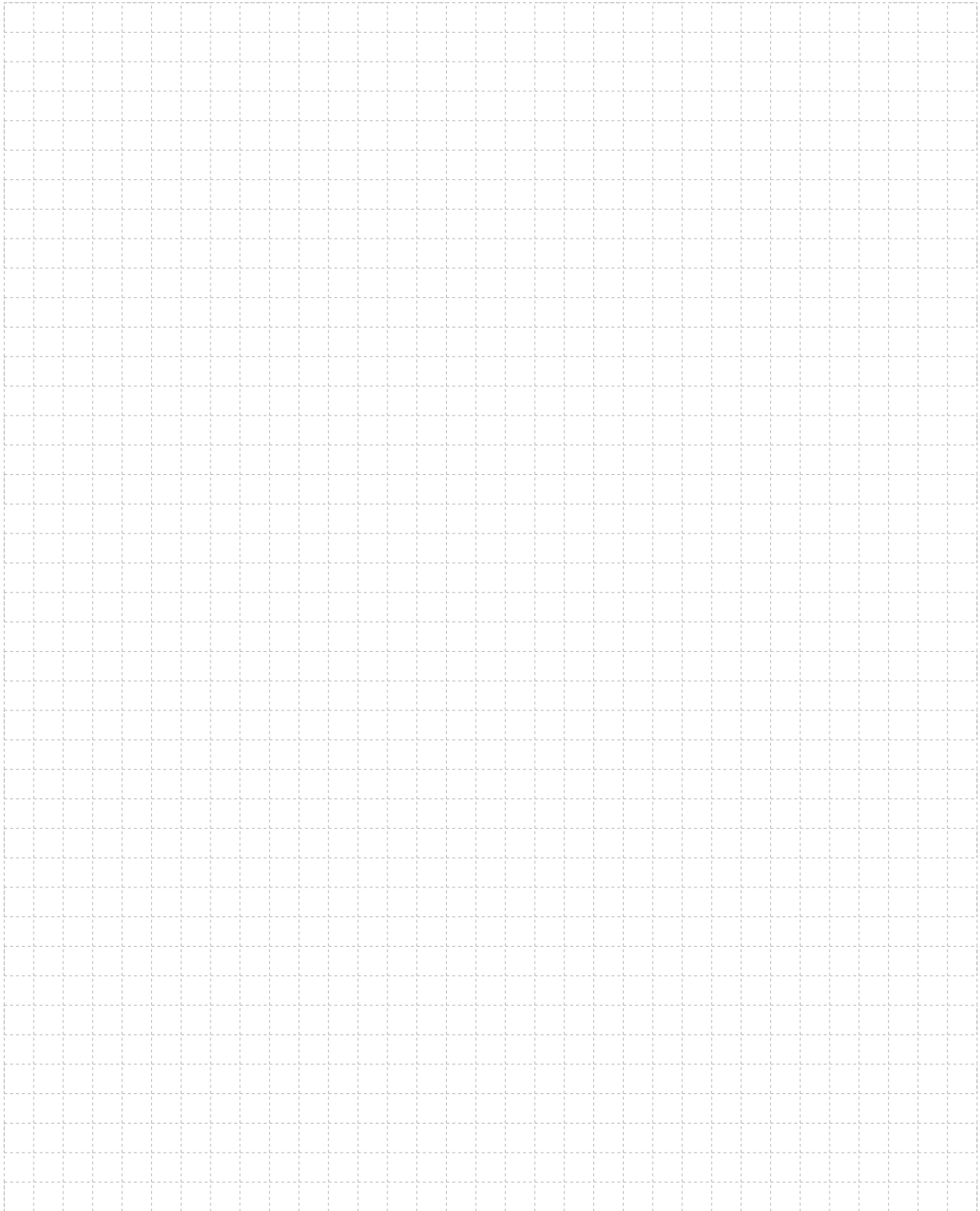


Name: \_\_\_\_\_

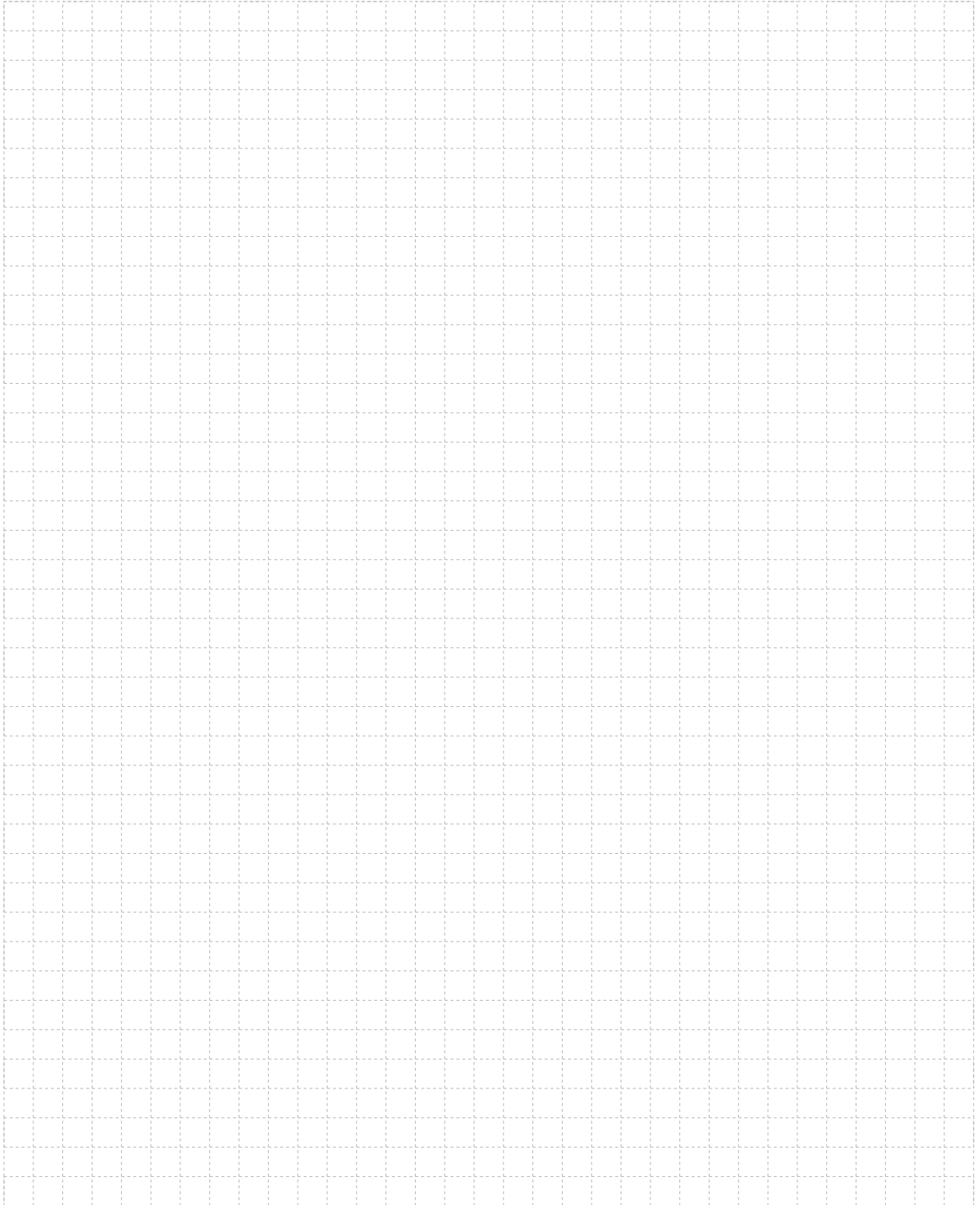
Matrikel-Nr.: \_\_\_\_\_

17

Ihre Lösung zu Aufgabe 8 (Fortsetzung):



Zusatzseiten für alle Aufgaben - Seite 1

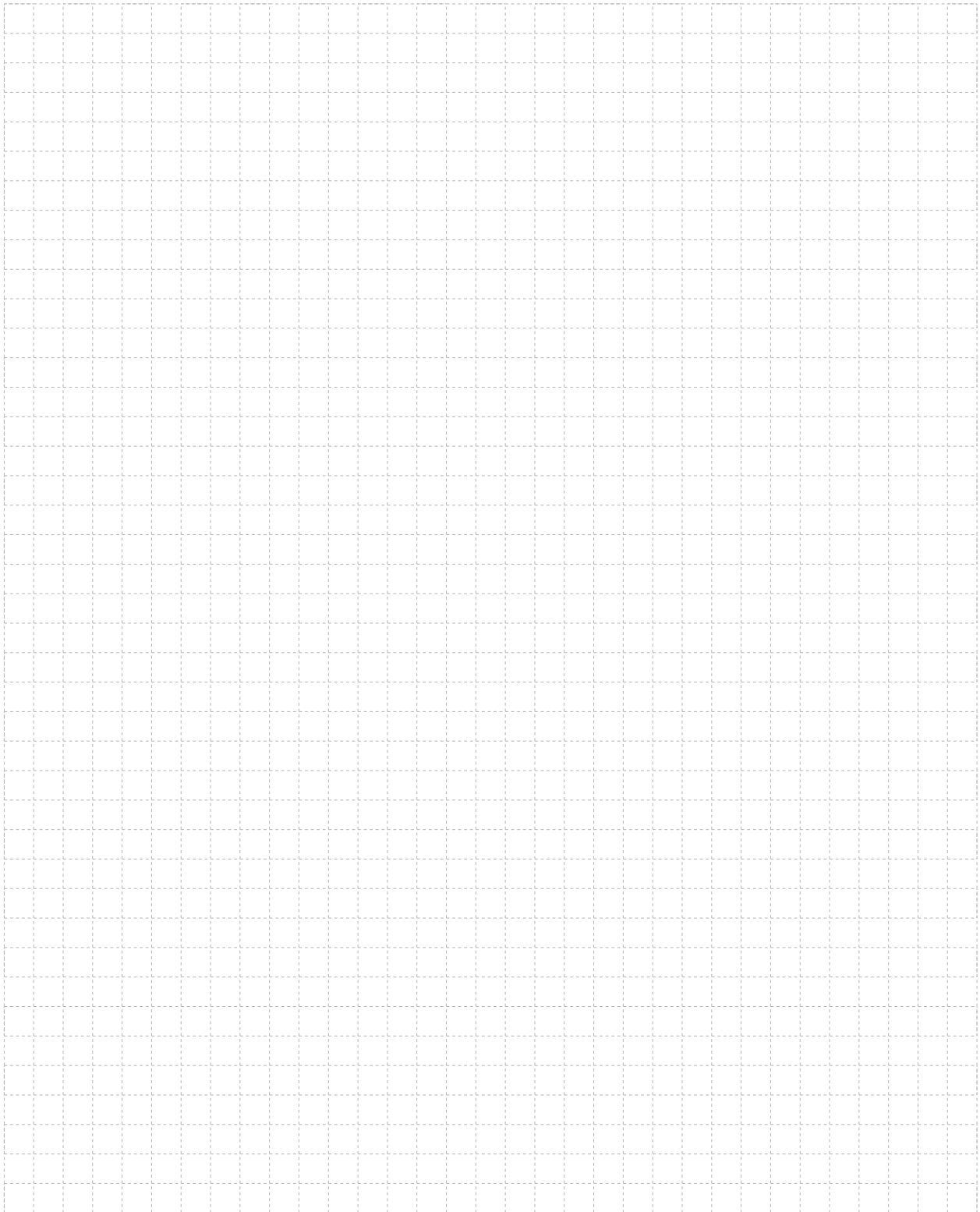


Name: \_\_\_\_\_

Matrikel-Nr.: \_\_\_\_\_

19

**Zusatzseiten für alle Aufgaben - Seite 2**



**Zusatzseiten für alle Aufgaben - Seite 3**

