



Programmierung von Systemen

im Sommersemester 2020

Institut für Softwaretechnik und Programmiersprachen
Prof. Dr. Matthias Tichy, Stefan Götz 10. August 2020, 8 Uhr

Bearbeitungszeit: 100 Minuten

Nachname:	Vorname:	Matrikelnummer:
Studiengang und Abschluss:		
TischNr:		
<p>Hiermit erkläre ich, dass ich prüfungsfähig bin. Sollte ich nicht auf der Liste der angemeldeten Studierenden aufgeführt sein, dann nehme ich hiermit zur Kenntnis, dass diese Prüfung nicht gewertet werden wird.</p> <hr/> <p>Datum, Unterschrift des Prüfungsteilnehmers</p>		

Zur allgemeinen Beachtung:

- Zur Bearbeitung der Klausur stehen **100 Minuten** zur Verfügung.
- Es sind keine zusätzlichen Hilfsmittel zugelassen.
- Sie können Ihre Lösungen **direkt auf die Aufgabenblätter** schreiben.
- Bei Erklärungsaufgaben werden ganze Sätze und nicht nur Stichpunkte erwartet.
- Falls Sie zusätzlich Papier benötigen:
 - Papier wird Ihnen zur Verfügung gestellt.
 - Schreiben Sie Ihren **Namen** und Ihre **Matrikelnummer** auf **jedes zusätzliche Blatt!**



Viel Erfolg!

Aufgabe	1	2	3	4	5	6	7	Σ
Max.	17.5	12.5	11.5	12	11	15	10	89.5
Punkte								
Hz.								

Note	Unterschrift Prüfer

Aufgabe 1 - OOP (6 + 4 + 3 + 3 + 1.5 = 17.5 Punkte)

Gegeben sei der Java-Code, welchen Sie auf dem beiliegenden Extrablatt finden können.

- a)** Zeichnen Sie ein Klassendiagramm für den gegebenen Java-Code. Achten Sie dabei auch auf Sichtbarkeiten für Attribute Assoziationen und Methoden.

- b) Implementieren sie die Methoden `transferComputingUnits(int number, Cluster to)` und `transferTo(Cluster to)` in den Klassen `Cluster` und `ComputingUnit`. Die Methode `transferComputingUnits` sollen die ersten `number` `ComputingUnits` aus dem aktuellen `Cluster` entfernen und dem übergebenen `Cluster` anhängen. Dabei sollen auch die `belongsTo` Referenzen via der Methode `transferTo` geändert werden. Sollten im aktuellen `Cluster` nicht genug `ComputingUnits` vorhanden sein sollen keine übertragen werden und `false` zurückgegeben werden, ansonsten `true`. Beachten Sie dabei, dass das manipulieren einer `Collection` während darüber iteriert wird undefiniertes Verhalten aufweist und deshalb vermieden werden muss!

```
public abstract class Cluster {  
    private final LinkedList<ComputingUnit> cus = new LinkedList<>();  
  
    public boolean transferComputingUnits(int number, Cluster to) {
```

```
    }  
}  
  
//continues on next page!
```

```
public class ComputingUnit {  
  private Cluster belongsTo;  
  private final int computingPower;  
  
  @Override  
  public void transferTo(Cluster to) {  
  
    }  
}
```

- c) Welche Ergänzungen an der Klasse `Computer` müssen vorgenommen werden um sicher zu stellen, dass nur maximal eine Instanz der Klasse erzeugt werden kann. Ergänzen Sie hierzu den folgenden Codeabschnitt:
-

```
public class Computer extends Cluster {
```

```
}
```

- d) Welche der folgenden Codestücke sind korrekt und welche führen zu **Compile-** oder **Laufzeitfehlern**? Begründen Sie, warum ein Codestück nicht korrekt ist. Sie können davon ausgehen dass die Aufrufe aus einer main-Methode in einem anderen *Package* gemacht werden welches die nötigen imports vorweist.
-

```
var computer = new Computer();  
System.out.println(computer.transferComputingUnits(100, new Cluster()));
```

```
Computer computer = new Computer();  
Cluster cluster = new Computer();  
var cu = new ComputingUnit(computer, 100);
```

```
ComputingUnit cu = new ComputingUnit();  
var computer = new Computer();  
cu.transferTo(computer);
```

- e) Der folgende Codeabschnitt verändert die Semantic von ComputingUnits grundlegend. Erklären Sie (z.B anhand eines Beispiels) was diese Änderung zur Folge hat und warum dies hier unerwünscht wäre.

```
public class ComputingUnit implements Transferable {  
    private Cluster belongsTo;  
    private static int computingPower;  
    public ComputingUnit(Cluster parent, int power) {  
        belongsTo = parent;  
        computingPower = power;  
    }  
}
```

Aufgabe 2 - JavaFX (2 + 2 + 2 + 2.5 + 4 = 12.5 Punkte)

Gegeben Sei der folgende Scenegraph.:

```

1 <BorderPane prefHeight="500.0" prefWidth="700.0"
2   xmlns="http://javafx.com/javafx/8.0.111" xmlns:fx="http://javafx.com/fxml/1">
3   <center>
4     <BorderPane>
5       <top>
6         <ToolBar>
7           <items>
8             <Button background-color="blue">
9               <graphic>
10                <ImageView pickOnBounds="true" preserveRatio="true">
11                  <image>
12                    <Image url="@/open.png" />
13                  </image>
14                </ImageView>
15              </graphic>
16            </Button>
17          </items>
18        </ToolBar>
19      </top>
20    <center>
21      <SplitPane orientation="VERTICAL">
22        <items>
23          <AnchorPane minHeight="0.0" minWidth="0.0">
24            <children>
25              <Label text="no file loaded..." />
26              <ScrollPane fx:id="imageScrollPane">
27                <content>
28                  <ImageView fx:id="imageView" />
29                </content>
30              </ScrollPane>
31            </children>
32          </AnchorPane>
33        </items>
34      </SplitPane>
35    </center>
36  </BorderPane>
37 </center>
38 </BorderPane>

```


- e) Welchen Code müssen Sie schreiben, damit beim Mausklick eines Buttons auf dem Button angezeigt wird ob dieser mit dem primären Maus-Button gedrückt wurde oder nicht?

Hinweis:

Verwenden Sie setzen des Textes die Methode `void setText(String text)` der Klasse `Button`.

```
1 public void start(Stage primaryStage) {
2     var sp = new StackPane();
3     var button = new Button();
4     sp.getChildren().add(button);
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28     primaryStage.setScene(new Scene(sp));
29     primaryStage.show();
30 }
```

Aufgabe 3 - Java IO (7 + 4.5 = 11.5 Punkte)

- a) Schreiben Sie eine Methode die die Inhalte von allen übergebenen Channels abwechselnd byteweise in die Datei **out.data** schreibt.

Achten Sie auf Behandlung der folgenden Exceptions mit einer jeweils passenden Meldung:

- `IOException`

Beachten Sie hierfür auch den JavaIO Teil des **CheatSheets** am Ende der Klausur.

Hinweis: Verwenden Sie die Methode `long getMaxSize(FileChannel... ics)` um die Länge des längsten channels zu bestimmen. Beachten Sie zudem

```
public static void combineBytes(FileChannel[] ics) {
```

```
}
```

- b)** Java bietet eine Vielzahl an Streams für die Verarbeitung von Daten an. Darunter auch gepufferte und un-gepufferte Varianten. Erläutern Sie anhand von **drei** sinnvoll gewählter Beispiele warum diese Vielzahl gerechtfertigt ist. Beachten Sie hierfür auch den Dateiein- und -Ausgabe Teil des **CheatSheets** am Ende der Klausur.

Aufgabe 4 - XML (2.5 + 8 + 1.5 = 12 Punkte)

Betrachten Sie das folgende XML-Dokument:

```

1 <xs:schema attributeFormDefault="unqualified"
2   elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="ComputingUnit">
4     <xs:complexType>
5       <xs:simpleContent>
6         <xs:extension base="xs:string">
7           <xs:attribute type="xs:byte" name="power" use="optional"/>
8         </xs:extension>
9       </xs:simpleContent>
10    </xs:complexType>
11  </xs:element>
12  <xs:element name="Location" type="xs:string"/>
13  </xs:element name="Computer">
14    <xs:complexType>
15      <xs:sequence>
16        <xs:element ref="ComputingUnit" maxOccurs="unbounded" minOccurs="0"/>
17      </xs:Sequence>
18      <xs:attribute type="xs:byte" name="id"/>
19    </xs:complexType>
20  </xs:element>
21  <xs:element name="Cluster">
22    <xs:complexType mixed="true">
23      <xs:sequence>
24        <xs:element ref="Location" minOccurs="1" maxOccurs="1">
25          <xs:element ref="Computer" minOccurs="0"/>
26          <xs:element ref="Cluster" minOccurs="0"/>
27        </xs:sequence>
28      </xs:complexType>
29    </xs:element>
30  </xs:schema>

```

- a) Das XML Dokument ist nicht wohlgeformt. Geben Sie die Zeilennummern der fünf vorkommenden Fehler an und erklären Sie wie diese korrigiert werden können. Konformität mit dem Schema für XSDs muss hierbei nicht berücksichtigt werden.

- b)** Die obige XML-Datei beschreibt ein XML-Schema. Schreiben Sie eine DTD-Datei so, dass diese die selben Dokumente validiert. Markieren Sie die Stellen an denen dies nicht möglich ist.
- c)** Erläutern Sie welchen XML-Parser Ansatz Sie verwenden würden um eine XML-Datei die dem obigen Schema entspricht in ein passendes JSON-Format zu transformieren.

Aufgabe 5 - ER-Modellierung (1.5 + 6 + 3.5 = 11 Punkte)

Ein Rechencluster besteht aus einer beliebigen Menge an Computern. Sowohl Computer als auch Rechencluster sind an einem bestimmten Ort aufgebaut. Computer können mehr als einem Cluster zugeordnet werden dafür müssen aber die Aufbauorte des Computers und der Cluster übereinstimmen. Damit Computer rechnen können werden mindestens 2 und maximal 42 ComputingUnits darin installiert welche nicht zwischen Computern geteilt werden können.

- a) Welche Anforderung aus dem obigen Sachverhalt können Sie nicht im **E-R-Diagramm** oder dem **relationalen Datenbankschema** modellieren und wo könnte diese stattdessen realisiert werden?

- b) Modellieren Sie den beschriebenen Sachverhalt als **E-R-Diagramm**.

- c) Erstellen Sie zu dem E-R-Diagramm das dazugehörige **relationale Datenbankschema**. Achten Sie hierbei auf sinnvoll gewählte **Primärschlüssel** und Stellen Sie sicher dass das Schema mindestens in 3. Normalform ist. Sie können bei Bedarf auch IDs einführen, um die Eindeutigkeit eines Primärschlüssels zu gewährleisten.

Aufgabe 6 - SQL (1 + 2.5 + 2 + 4.5 + 4 + 1 = 15 Punkte)

Gegeben seien die folgenden Relationenschemata (auch zu finden auf dem beiliegenden Extrablatt):

Schiffe		
<u>SID</u>	Name	MID

Rennen			
<u>RID</u>	RennName	Startzeit	Preisgeld

Schiffsrennen	
<u>SID</u>	<u>RID</u>

Matrosen	
<u>MID</u>	MName

Besatzung	
<u>MID</u>	<u>SID</u>

Primärschlüssel sind unterstrichen, Fremdschlüssel sind **fett** dargestellt. Formulieren Sie, insofern nicht anders spezifiziert, die **SQL-Statements** zur Lösung folgender Teilaufgaben:

- a)** Setzen Sie das Preisgeld des Rennens 'Segl101' auf 1000.

- b)** Geben Sie an, wieviele Schiffe existieren bei denen 'Titanic' im Namen vorkommt.

- c) Übersetzen Sie den folgenden Relationenalgebra-Ausdruck nach SQL.
- $$\pi_{Preisgeld}(\sigma_{(Name=Blitzkuigel)}(Rennen \bowtie Schiffsrennen \bowtie Schiffe))$$

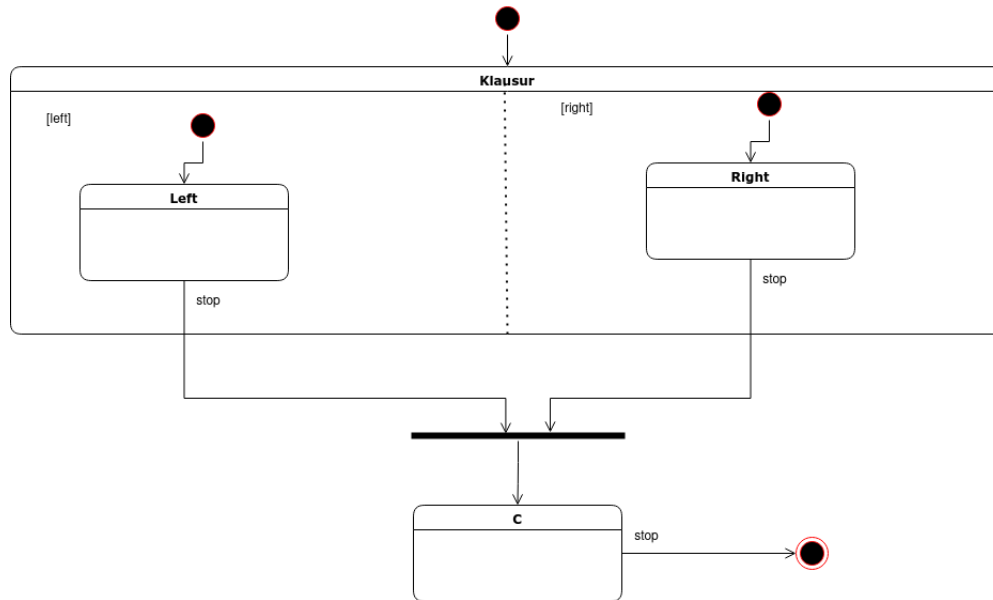
d) Geben Sie die Namen aller Schiffe aus, die mehr als 3 Besatzungsmitglieder haben. Besatzungen werden mittels der Tabelle *Besatzungen* die zwischen die Schiffe und Matrosen Tabelle gestellt ist definiert.

e) Geben Sie die Namen aller der Schiffe an, deren Besitzer nicht Teil der Besatzung ist.
Hiweis: Besitzer sind als MID in den Schiffen vermerkt!

f) Geben Sie die Namen aller Matrosen aus, die mindestens ein Schiff besitzen.

Aufgabe 7 - Threads (4 + 2 + 2 + 2 = 10 Punkte)

a) Betrachten Sie den folgenden Zustandsautomaten:



Ergänzen Sie den folgenden Code so, dass er den im Zustandsautomaten abgebildeten Ablauf darstellt. Sie können hierbei davon ausgehen, dass die Klassen `Left`, `Right` und `C` die internen Abläufe der linken bzw. rechten Seite des `Klausur` Zustandes und des `C` Zustandes abbilden. Um deren Implementierung muss sich nicht gekümmert werden.

```

1 public class StateManager {
2     public static void main(String... args) throws Exception{
3         Thread left  = new Left();
4         Thread right = new Right();
5         Thread c     = new C();
6
7
8
9
10
11
12
13
14         System.out.println("end");
15     }
16 }
  
```

- b)** Gegeben Sei der folgende Codeabschnitt welcher nicht immer die erwarteten 10000 ausgibt. Erklären Sie, mit Nennung der relevanten Zeilen warum dies nicht der Fall ist und Skizzieren Sie, wieder mit Nennung der relevanten Zeilen, zwei Arten um den Ablauf des Programmes Threadsicher zu machen.

Hinweis: Mehrfachnennungen des selben Prinzips sind nicht erlaubt.

```
1 public class MyThreads extends Thread {
2     public static int counter = 0;
3
4     public static void main(String... strings) throws Exception {
5         var myThreads = new LinkedList<Thread>();
6         for (int i = 0; i < 100; i++) {
7             mt.start();
8         }
9         System.out.println("counter = " + counter);
10    }
11
12    @Override
13    public void run() {
14        for (int i = 0; i < 100; i++) {
15            counter++;
16        }
17    }
18 }
```

- c)** Erläutern Sie warum es als schwierig erachtet wird Deadlocks in Code auszuschließen. Gehen Sie hierbei auch darauf ein wieso Deadlocks überhaupt entstehen.

- d)** Im Kontext des sogenannten ‘Actor-Modells’ gibt es die namensgebenden Aktoren. Was zeichnet diese aus?

Aufgabe 1 - OOP

```

1
2 public class ComputingUnit implements Transferable {
3     private Cluster belongsTo;
4     private final int computingPower;
5
6     public ComputingUnit(Cluster parent, int power) {
7         belongsTo = parent;
8         computingPower = power;
9     }
10
11     @Override
12     public void transferTo(Cluster to) {
13         // TODO Auto-generated method stub
14     }
15 }
16
17 public abstract class Cluster {
18     private final LinkedList<ComputingUnit> cus = new LinkedList<>();
19
20     public boolean transferComputingUnits(int number, Cluster to) {
21         // TODO Auto-generated method stub
22         return false;
23     }
24
25 }
26
27 public interface Transferable {
28     public void transferTo(Cluster to);
29 }
30
31 public class Computer extends Cluster {
32
33 }

```

Aufgabe 6 - SQL

Schiffe		
<u>SID</u>	Name	MID

Rennen			
<u>RID</u>	RennName	Startzeit	Preisgeld

Schiffsrennen	
<u>SID</u>	<u>RID</u>

Matrosen	
<u>MID</u>	MName

Besatzung	
<u>MID</u>	<u>SID</u>

JavaIO

FileOutputStream

Modifier and Type	Method and Description
void	close() Closes this file output stream and releases any system resources associated with this stream.
protected void	finalize() Cleans up the connection to the file, and ensures that the <code>close</code> method of this file output stream is called when there are no more references to this stream.
FileChannel	getChannel() Returns the unique FileChannel object associated with this file output stream.
FileDescriptor	getFD() Returns the file descriptor associated with this stream.
void	write(byte[] b) Writes <code>b.length</code> bytes from the specified byte array to this file output stream.
void	write(byte[] b, int off, int len) Writes <code>len</code> bytes from the specified byte array starting at offset <code>off</code> to this file output stream.
void	write(int b) Writes the specified byte to this file output stream.

JavaIO

FileChannel

Modifier and Type	Method and Description
abstract <code>MappedByteBuffer</code>	<code>map(FileChannel.MapMode mode, long position, long size)</code> Maps a region of this channel's file directly into memory.
static <code>FileChannel</code>	<code>open(Path path, OpenOption... options)</code> Opens or creates a file, returning a file channel to access the file.
static <code>FileChannel</code>	<code>open(Path path, Set<? extends OpenOption> options, FileAttribute<?>... attrs)</code> Opens or creates a file, returning a file channel to access the file.
abstract <code>long</code>	<code>position()</code> Returns this channel's file position.
abstract <code>FileChannel</code>	<code>position(long newPosition)</code> Sets this channel's file position.
abstract <code>int</code>	<code>read(ByteBuffer dst)</code> Reads a sequence of bytes from this channel into the given buffer.
<code>long</code>	<code>read(ByteBuffer[] dsts)</code> Reads a sequence of bytes from this channel into the given buffers.
abstract <code>long</code>	<code>read(ByteBuffer[] dsts, int offset, int length)</code> Reads a sequence of bytes from this channel into a subsequence of the given buffers.
abstract <code>int</code>	<code>read(ByteBuffer dst, long position)</code> Reads a sequence of bytes from this channel into the given buffer, starting at the given file position.
abstract <code>long</code>	<code>size()</code> Returns the current size of this channel's file.
abstract <code>long</code>	<code>transferFrom(ReadableByteChannel src, long position, long count)</code> Transfers bytes into this channel's file from the given readable byte channel.
abstract <code>long</code>	<code>transferTo(long position, long count, WritableByteChannel target)</code> Transfers bytes from this channel's file to the given writable byte channel.
abstract <code>FileChannel</code>	<code>truncate(long size)</code> Truncates this channel's file to the given size.
<code>FileLock</code>	<code>tryLock()</code> Attempts to acquire an exclusive lock on this channel's file.
abstract <code>FileLock</code>	<code>tryLock(long position, long size, boolean shared)</code> Attempts to acquire a lock on the given region of this channel's file.
abstract <code>int</code>	<code>write(ByteBuffer src)</code> Writes a sequence of bytes to this channel from the given buffer.
<code>long</code>	<code>write(ByteBuffer[] srcs)</code> Writes a sequence of bytes to this channel from the given buffers.
abstract <code>long</code>	<code>write(ByteBuffer[] srcs, int offset, int length)</code> Writes a sequence of bytes to this channel from a subsequence of the given buffers.
abstract <code>int</code>	<code>write(ByteBuffer src, long position)</code> Writes a sequence of bytes to this channel from the given buffer, starting at the given file position.

There are two hard things in computer science: cache invalidation, naming things, and off-by-one errors.

Datei- und -ausgabe

