

Klausur Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

Hinweise

- Bevor Sie mit der Bearbeitung der Aufgaben beginnen, müssen Sie auf allen Blättern Ihren Namen und Ihre Matrikelnummer eintragen. Dafür bekommen Sie zusätzlich fünf Minuten Zeit.
- Der Klausurtext enthält ausreichend Platz zur Lösung der Aufgaben. Sie können auch die Rückseiten der Blätter für Ihre Lösungen nutzen. Sofern Sie zusätzliches Papier benötigen, wenden Sie sich an die Aufsicht. Die Nutzung eigenen Papiers ist nicht gestattet.
- Sollte Ihre Lösung nicht unmittelbar unter oder neben der Aufgabenstellung stehen, machen Sie bitte einen entsprechenden Hinweis. Streichen Sie diejenigen Teile der von Ihnen geschriebenen Texte deutlich durch, die **nicht** in die Bewertung eingehen sollen.
- Die Klausur ist zusammengeheftet. Die Heftung darf nicht geöffnet werden.
- Die Aufsicht gibt Ihnen keine Hilfestellung beim Lösen der Aufgaben.
- Teilnehmer, die eine Einzelklausur über **ein Teilgebiet** schreiben, müssen zum Bestehen **mindestens 20 Punkte** erreichen. Maximal können 50 Punkte erreicht werden.
- Teilnehmer, die die Gesamtklausur über **beide Teilgebiete** schreiben, müssen zum Bestehen in beiden Teilgebieten **jeweils mindestens 20 Punkte** erreichen. Maximal können jeweils 50 Punkte erreicht werden.

Viel Erfolg!

Aufgabe	Typ	a) b) c) d) e) f)	erreichbare Punkte	erreichte Punkte
1	Multiple Choice		7	
2	Graphen	1 1 4 1 2	9	
3	Datenstruktur	7 4 2 1 2	16	
4	Suchbäume	3 2 1	6	
5	Definitionen		12	
Summe			50	

Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

Aufgabe 1: Multiple Choice-Test

Hinweise:

Zu jeder Frage gibt es **genau eine** richtige Antwort.

Bewertung je Frage:

Richtiges Kreuz: 1 Punkt

Falsches Kreuz: 0,5 Punkte Abzug

Mehr als ein Kreuz: 0 Punkte

Maximal zu erreichende Gesamtpunktzahl: 7 Punkte

Minimal zu erreichende Gesamtpunktzahl: 0 Punkte

Sollten Sie Ihre Antwort korrigieren, machen Sie bitte **unmissverständlich** deutlich, welche Antwort gilt, andernfalls wird diese Frage nicht bewertet.

- Ein Binärbaum werde nach der üblichen in der Vorlesung vorgestellten Methode unter Verwendung von Zeigern auf Nachfolgeknoten implementiert. Wie viele solcher Zeiger (einschließlich Zeiger auf `null`) müssen zur Speicherung von N Knoten im Binärbaum verwendet werden?
 - N-1
 - N
 - 2N
- Welche der folgenden Aussagen gilt für jeden Baum, der ein Heap ist?
 - Alle Elemente im linken Unterbaum der Wurzel haben eine kleinere Beschriftung als die Wurzel, und alle Elemente im rechten Unterbaum der Wurzel haben eine größere Beschriftung als die Wurzel.
 - Die Wurzel des Baumes ist der Knoten mit der kleinsten Beschriftung.
 - Für beliebige Knoten a und b des Baumes gilt: Ist die Beschriftung von a kleiner als die Beschriftung von b , so muss die Tiefe des Knotens a kleiner als die Tiefe des Knotens b sein
- Welche Aussage über abstrakte Klassen ist stets wahr?
 - Eine abstrakte Klasse enthält ausschließlich abstrakte Methoden.
 - eine von einer abstrakten Klasse abgeleitete Unterklasse muss alle geerbten abstrakten Methoden implementieren oder selbst als abstrakt definiert werden.
 - Von einer abstrakten Klasse gebildete Instanzen sind immer abstrakt.

Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

- Welche Aussage über Interfaces in der Sprache Java ist **falsch**?
 - Wenn eine Klasse ein Interface implementiert, muss diese Klasse abstrakt sein.
 - Ein Interface kann von mehreren anderen Interfaces erben.
 - Die Attribute eines Interfaces sind immer Konstanten.

- Sei G ein beliebiger zusammenhängender gerichteter Graph, der einen Zyklus enthält. Welche der folgenden Aussagen ist dann **falsch**?
 - Jeder Spannbaum von G hat weniger Kanten als G .
 - Der Spannbaum von G ist eindeutig bestimmt.
 - G ist kein Spannbaum.

- Ein Zyklus in einem gerichteten Graphen ist definiert als
 - Weg von einem Knoten zu sich selbst.
 - einfacher Weg von einem Knoten zu sich selbst.
 - Semiweg von einem Knoten zu sich selbst.

- Der Speicherbedarf von Adjazenzmatrizen von nicht attribuierten Graphen ist
 - abhängig von der Zahl der Kanten des Graphen, aber nicht von der Zahl der Knoten
 - abhängig von der Zahl der Knoten des Graphen, aber nicht von der Zahl der Kanten
 - abhängig sowohl von der Zahl der Knoten des Graphen als auch von der Zahl der Kanten

**Klausur - Teilgebiet: Programmierung und
Programmiersprachen**

Vor- und Nachname: _____ Mat.-Nr.: _____

Aufgabe 2: Graphen

a) Geben Sie die formale Definition eines gerichteten Graphen an. **(2 Punkte)**

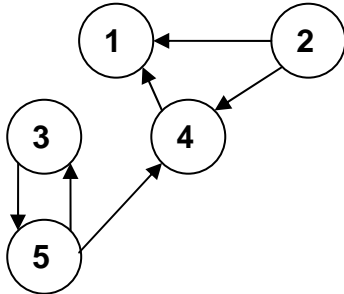
b) Zeichnen Sie einen gerichteten Graphen, der schwach zusammenhängend, aber nicht stark zusammenhängend ist. **(1 Punkt)**

c) Für die Speicherung von Graphen haben Sie Adjazenzlisten und Adjazenzmatrizen kennengelernt. Beschreiben Sie kurz, welchen Vorteil eine Darstellung als Adjazenzliste und welchen Vorteil eine Darstellung als Adjazenzmatrix hat. Wann ist eine Adjazenzliste, wann eine Adjazenzmatrix zu bevorzugen? **(4 Punkte)**

Klausur - Teilgebiet: Programmierung und Programmiersprachen

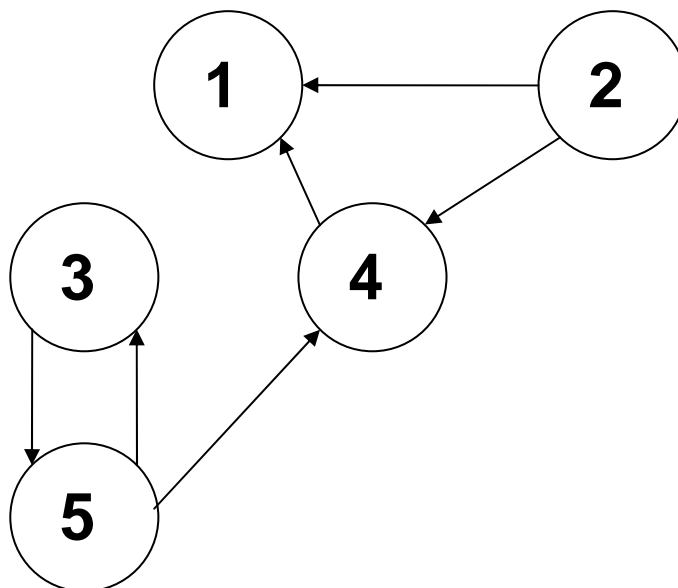
Vor- und Nachname: _____ Mat.-Nr.: _____

d) Geben Sie die Adjazenzmatrix für den im folgenden Bild dargestellten gerichteten Graphen G an. **(1 Punkt)**



	1	2	3	4	5
1					
2					
3					
4					
5					

e) Ergänzen Sie die graphische Darstellung von G so, dass der dargestellte Graph der transitiven Hülle von G entspricht. **(2 Punkte)**



Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

Aufgabe 3: Datenstruktur

Gegeben seien die folgenden Java-Codefragmente:

```
class Knoten {
    Object inhalt;
    Knoten links;
    Knoten rechts;
    Knoten (Object e1) {
        this.inhalt = e1;
    }
}

class BinaerBaum{
    private Knoten wurzel;
    private Knoten cursor;
    boolean isEmpty() {
        return wurzel == null;
    }
}
```

Weiterhin sei eine Implementierung der Klasse Queue mit folgenden Methoden gegeben:

```
class Queue () {

    /** isEmpty testet, ob diese Queue leer ist. */
    boolean isEmpty() {...}

    /** peek gibt das erste Element der Queue zurück, ohne es
        aus der Queue zu entfernen */
    Knoten peek() {...}

    /** poll gibt das erste Element der Queue zurück, und entfernt es
        aus der Queue. Gibt null zurück, falls die Queue leer ist. */
    Knoten poll() {...}

    /** offer fügt ein Element in die Queue ein. */
    void put (Knoten knoten) {...}
}
```

Bei den folgenden Aufgaben reicht es, die neu hinzukommenden Codeabschnitte zu notieren; der bereits gegebene Code muss nicht wiederholt werden.

Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

a) Durch das folgende Interface ist eine Methode `breitensuche(Knoten knoten)` definiert, die einen Binärbaum unter Verwendung der Breitensuche nach einem gegebenen Knoten durchsucht und diesen als Ergebnis zurückgibt.

```
public interface Searchable {  
    public Knoten breitensuche(Knoten knoten)  
                                   throws NoSuchElementException  
}
```

Notieren Sie für die Klasse `BinaerBaum` den Code, der nötig ist, um das Interface `Searchable` unter Verwendung der Klasse `Queue` zu implementieren! **(7 Punkte)**

Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

b) Ergänzen Sie die Klasse `Knoten` um eine Methode, mit der Objekte der Klasse `Knoten` auf Gleichheit des Attributes `inhalt` getestet werden können. Überschreiben Sie hierzu die Methode `equals(Object object)`! **(4 Punkte)**

c) Worin unterscheiden sich die für Vergleiche in Java verwendeten Methoden `equals(Object object)` und `compareTo(Object object)` voneinander? **(2 Punkte)**

d) Begründen Sie, welche unerwünschten Effekte auftreten können, wenn die Methode `breitensuche(Knoten knoten)` der Klasse `Knoten` in mehreren parallel laufenden Threads verwendet werden soll? **(1 Punkt)**

e) Welche Maßnahmen können bei der Implementierung der Methode `breitensuche(Knoten knoten)` getroffen werden, um sicherzustellen, dass bei Ausführung mehrerer parallel laufender Threads keine unerwünschten Effekte auftreten können? **(2 Punkte)**

Klausur - Teilgebiet: Programmierung und Programmiersprachen

Vor- und Nachname: _____ Mat.-Nr.: _____

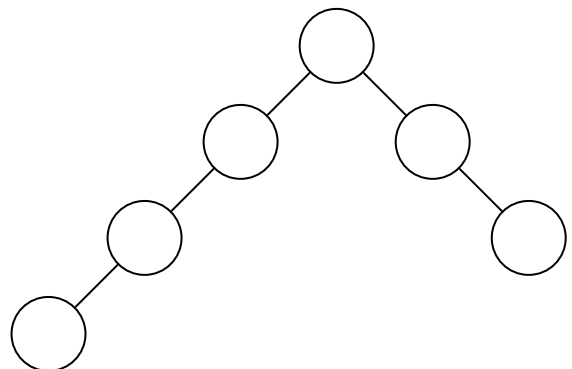
Aufgabe 4: Binäre Suchbäume

a) Benutzen Sie den in der Vorlesung behandelten Algorithmus, um durch wiederholtes Einfügen in einen leeren Baum einen binären Suchbaum zu erzeugen. Fügen Sie dabei die folgenden Zahlen in der angegebenen Reihenfolge ein und zeichnen Sie den entstehenden Suchbaum. **(3 Punkte)**

16, 11, 3, 14, 20, 18, 15, 19, 4

b) Unter welchen Voraussetzungen ergibt sich durch wiederholtes Einfügen von Elementen e_1, e_2, \dots, e_n in einen leeren binären Suchbaum ein „entarteter“ Baum, der die in der nebenstehenden Abbildung gezeigte Form hat?

(2 Punkte)



c) Unter welcher Voraussetzung sind für eine erfolglose Suche in einem binären Suchbaum mit n Knoten n Vergleichsoperationen nötig? **(1 Punkt)**

**Klausur - Teilgebiet: Programmierung und
Programmiersprachen**

Vor- und Nachname: _____ Mat.-Nr.: _____

Aufgabe 5: Vermischte Definitionen

a) Was verstehen Sie unter einem abstrakten Datentyp? **(1 Punkt)**

b) Der abstrakte Datentyp Stack hat die Operationen isEmpty, pop, push und top. Geben Sie die Signatur der Operation isEmpty an. **(1 Punkt)**

c) Was ist ein Konstruktor? **(1 Punkt)**

d) Was verstehen Sie unter Persistenz? **(1 Punkt)**

e) Wozu dienen Streams in der Sprache Java? **(1 Punkt)**

**Klausur - Teilgebiet: Programmierung und
Programmiersprachen**

Vor- und Nachname: _____ Mat.-Nr.: _____

f) Definieren Sie im Multithreading-Kontext die Begriffe kritischer Abschnitt, Monitor und Deadlock. **(3 Punkte)**

g) Erläutern Sie kurz den Unterschied zwischen einer Klasse und einem Objekt. **(2 Punkte)**

h) Erläutern Sie kurz den Unterschied zwischen abstrakten Klassen und Interfaces in der Sprache Java. **(2 Punkte)**