

Vorname

Nachname

Matrikel-Nr

Diese Klausur ist mein **letzter Prüfungsversuch** (bitte ankreuzen):      Ja ☐      Nein ☐

Schreiben Sie jede Lösung auf die Vorderseite eines **neuen Blattes** (und lassen Sie die Rückseiten Ihrer Lösungsblätter **leer**). Die Aufgaben 4 bis 6 stehen auf der Rückseite dieses Blatts!

### Aufgabe 1 (25 Punkte): Zwei voneinander unabhängige Teilaufgaben.

Beide müssen **rekursiv** gelöst werden (sie dürfen keine Schleifen enthalten).

```

1  // -----
2  // Teilaufgabe 1.1.: Schreiben Sie eine rekursive Methode, die der
3  // folgenden Spezifikation entspricht und die darunter stehende Bedingung
4  // erfuehlt:
5
6  static public int add(int m, int n) {
7      // Verlaesst sich darauf, dass m und n nicht negativ sind.
8      // Liefert die Summe von m und n.
9      ...
10 } // add
11
12 // Die rekursive Methode add muss folgende Bedingung erfuehllen:
13 // 1. Die Operatoren + und - werden darin nicht benutzt.
14 // 2. Statt dessen werden die folgenden beiden Methoden aufgerufen:
15
16 static public int plus1 (int n) {return ++n;}
17 static public int minus1(int n) {if (n>0) return --n; else return 0;}
18
19 // -----
20 // Teilaufgabe 1.2.: Schreiben Sie eine rekursive Methode, die der
21 // folgenden Spezifikation entspricht:
22
23 static public String als12erZahl(int n) {
24     // Liefert n als 12-er-Zahl (als Zahl im Zahlensystem mit
25     // der Basis 12), negative Zahlen mit einem Minuszeichen - davor.
26     // Beispiele:
27     // als12erZahl( 11) ist gleich "B"
28     // als12erZahl(-11) ist gleich "-B"
29     // als12erZahl( 12) ist gleich "10"
30     // als12erZahl(-12) ist gleich "-10"
31     // als12erZahl( 13) ist gleich "11"
32     // als12erZahl( 22) ist gleich "1A"
33     // als12erZahl(130) ist gleich "AA"
34     //
35     // Tip: Wenn Sie die int-Werte 0, 1, ..., 9, 10, 11 in
36     // die Strings "0", "1", ..., "9", "A", "B" umwandeln wollen,
37     // dann geht das besonders einfach mit einer Reihung von Strings.
38     ...
39 } // als12erZahl

```

### Aufgabe 2 (15 Punkte): Die Klasse ArrayList<K> implementiert die Schnittstelle List<K>.

Die folgende Klasse ArraySet<K> soll die Schnittstelle Set<K> implementieren. Sie brauchen nur noch die beiden add-Methoden (mit einem bzw. zwei Parametern) so zu vervollständigen, dass alle Bedingungen der Schnittstelle Set<K> erfüllt werden (auch die weichen Bedingungen).

```

1  class ArraySet<K> extends ArrayList<K> implements Set<K> {
2      @Override
3      public boolean add(K k) { ... }
4
5      @Override
6      public void add(int index, K k) { ... }
7  } // class ArraySet<K>

```

**Aufgabe 3 (15 Punkte):** Es folgen hier 5 .xml-Dateien und (unten rechts) eine .dtd-Datei:

<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE zitat SYSTEM "Zitat.dtd"&gt; &lt;!-- Datei ZitatA.xml --&gt; &lt;zitat&gt;   &lt;autor&gt;C. Morgenstern&lt;/autor&gt;   &lt;titel&gt;Das Mondschaft&lt;/titel&gt;   &lt;text nr="z1" laenge="2Z"&gt;     Das Mondschaft steht auf     weiter Flur   &lt;/text&gt;   &lt;text nr="z2" laenge="2Z"&gt;     Das Mondschaft spricht zu sich     im Traum   &lt;/text&gt; &lt;/zitatz&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE zitat SYSTEM "Zitat.dtd"&gt; &lt;!-- Datei ZitatB.xml --&gt; &lt;zitat&gt;   &lt;autor land="DE" zeit="19JH"&gt;     J. W. von Goethe   &lt;/autor&gt;   &lt;titel&gt;Faus&lt;/titel&gt;   &lt;text nr="dreier" laenge="1Z"&gt;     Vom Eise befreit sind     Strom und Bäche   &lt;/text&gt;   &lt;text nr="z2"&gt;     Heinrich!     Mir graut's vor dir.   &lt;/text&gt;   &lt;text nr="z3"/&gt; &lt;/zitatz&gt; </pre>
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE zitat SYSTEM "Zitat.dtd"&gt; &lt;!-- Datei ZitatC.xml --&gt; &lt;zitat zeit="20JH"&gt;   &lt;autor land="DE"&gt;Beatles&lt;/autor&gt;   &lt;titel&gt;     Sgt. Pepper's     Lonely Hearts     Club Band   &lt;/titel&gt;   &lt;text nr="z2" laenge="2Z"&gt;     It was twenty years ago today,     Sgt. Pepper taught     the band to play   &lt;/text&gt;   &lt;text nr="z1" laenge="1Z"&gt;     Lucy in the sky with diamonds   &lt;/text&gt; &lt;/zitatz&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE zitat SYSTEM "Zitat.dtd"&gt; &lt;!-- Datei ZitatD.xml --&gt; &lt;zitat&gt;   &lt;autor land="DE"&gt;     J. Gosling   &lt;/autor&gt;   &lt;titel&gt;     The Java Language Specification   &lt;/titel&gt;   &lt;text nr="xkr7" laenge="1Z"&gt;     The Java programming language     is a strongly typed language   &lt;/text&gt;   &lt;text nr="yypft33"&gt;     The Java programming language     is related to C but ...   &lt;/text&gt;   &lt;text/&gt; &lt;/zitatz&gt; </pre>
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE zitat SYSTEM "Zitat.dtd"&gt; &lt;!-- Datei ZitatE.xml --&gt; &lt;zitat&gt;   &lt;titel&gt;Unbekannt&lt;/titel&gt;   &lt;text nr="z1" laenge="1Z"&gt;     Wenn es schief gehen kann,     wird es schief gehen!   &lt;/text&gt;   &lt;text nr="z1" laenge="1Z"&gt;     Nur die paranoiden überleben!   &lt;/text&gt; &lt;/zitatz&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!-- Datei Zitat.dtd --&gt; &lt;!ELEMENT zitatz (autor?, titel, text+)&gt; &lt;!ELEMENT autor (#PCDATA)&gt; &lt;!ELEMENT titel (#PCDATA)&gt; &lt;!ELEMENT text (#PCDATA)&gt;  &lt;!ATTLIST autor   land (DE EN ES FR XX) #REQUIRED   zeit (alt 19JH 20JH neu) "neu" &gt;  &lt;!ATTLIST text   nr ID #REQUIRED   laenge CDATA #IMPLIED &gt; </pre>

Welche der .xml-Dateien sind gültig? Und welche sind ungültig? Geben Sie für jede ungültige Datei (möglichst kurz) den *Grund* ihrer Ungültigkeit an.

**Aufgabe 4 (15 Punkte):** Stellen Sie die folgenden vier Variablen `rd`, `ri`, `st` und `ro` als Bojen dar (nach Ihrer Wahl in vereinfachter oder in ausführlicher Bojendarstellung):

```

1      double[] rd = {};
2      int[]    ri = {10, 20};
3      String   st = "Hallo";
4
5      Object[] ro = {rd, ri, st, "Hallo", null};

```

**Aufgabe 5 (15 Punkte):** Schreiben Sie eine Methode, die der folgenden Spezifikation entspricht:

```

1  static public void aufgabe5(Object ob)
2      throws IllegalAccessException, InvocationTargetException
3  {
4      // Wenn ob eine GEEIGNETE add-Funktion enthaelt, dann wird diese
5      // mit den Parametern 3 und 4 aufgerufen und ihr Ergebnis wird
6      // (mit pln) ausgegeben.
7      // Wenn ob keine GEEIGNETE Funktion enthaelt, wird die
8      // Fehlermeldung "Keine add-Methode!" ausgegeben.
9      // GEEIGNET ist eine Funktion, wenn sie folgende Eigenschaften hat:
10     //
11     // 1. Sie hat den Namen add
12     // 2. Sie ist eine oeffentliche Klassenmethode
13     // 3. Sie hat den Rueckgabebetyp int
14     // 4. Sie hat genau 2 Parameter
15     // 5. Diese Parameter sind beide vom Typ int
16     //
17     // Hinweis: Beim Schreiben der Methode brauchen Sie Ausnahmen nicht
18     // zu beruecksichtigen (das erledigt die throws-Klausel bereits).
19     ...
20 } // aufgabe5

```

**Kleiner Tip:**  
`(getName())`.  
`(getReturnType())`  
`(getParameterTypes())`

**Aufgabe 6: (15 Punkte)** Beantworten Sie die folgenden Fragen möglichst *kurz*, aber *genau* und benutzen Sie dabei die im seminaristischen Unterricht behandelten *Fachbegriffe*:

1. Was ist besser an *Fäden* (engl. threads) als an *Prozessen*?
2. Angenommen, ein Algorithmus A1 hat eine Zeikomplexität von  $O(n^3)$  und braucht zu Lösung eines Problems der Größe 100 etwa 500 Schritte. Wie viele Schritte braucht A1 dann zur Lösung eines Problems der Größe 200?
3. *Wie viele Schritte* braucht man höchstens, um in einer *sortierten Reihung* der Länge 4 Millionen *in einem negativen Fall* zu suchen? Geben Sie die konkrete ("ausgerechnete") Zahl an, keine Formel.
4. *Welche Tiefe* muss ein binärer Baum, der 8 Millionen Knoten enthält, mindestens haben?
5. Wenn man einen binären Baum als Reihung `rei` darstellt, welche Indizes haben dann die beiden *Nachfolger* des Knotens `rei[7]`? Und welchen Index hat der *Vorgänger* von `rei[7]`?
6. Welche der folgenden Worte bezeichnen (notwendige oder optionale) Teile eines *Umwandlungsbefehls* im *Formatstring* eines `format`-Befehls?  
Länge, Fahrkarte, Schalter, Präzision, Breite, Höhe, Tiefe, Genauigkeit, UZiffer, VZiffer, UBuchstabe, VBuchstabe, Index, Schleife
7. Angenommen, die Variable `kob` zeigt auf ein `Class`-Objekt.  
Was liefert dann der Funktionsaufruf `kob.getDeclaredFields()`?
8. *Sortierte Reihungen* werden manchmal als "Ersatz für Sammlungen" verwendet. Nennen Sie zwei Nachteile, die sortierte Reihungen in einem solchen Fall haben.

**Beurteilung dieser Klausur:**

A1	
A2	
A3	
A4	
A5	
A6	
Summe	
Note	
Datum	

**Korrigierte Beurteilung:**

A1	
A2	
A3	
A4	
A5	
A6	
Summe	
Note	
Datum	

**Lösung 1 (25 Punkte):** Zwei voneinander unabhängige Teilaufgaben. Beide müssen rekursiv gelöst werden (sie dürfen keine Schleifen enthalten).

```

1  // Teilaufgabe 1.1.: Schreiben Sie eine rekursive Methode, die der
2  // folgenden Spezifikation entspricht und die darunter stehenden Bedingungen
3  // erfuehlt:
4
5  static public int add(int m, int n) {
6      // Verlaesst sich darauf, dass m und n nicht negativ sind.
7      // Liefert die Summe von m und n.
8
9      if (n == 0) {
10         return m; // m + 0 = m
11     } else {
12         return plus1(add(m, minus1(n))); // m + n = (m + (n-1)) + 1
13     }
14 } // add

```

```

1  // Teilaufgabe 1.2.: Schreiben Sie eine rekursive Methode, die der
2  // folgenden Spezifikation entspricht:
3
4  static public String als12erZahl(int n) {
5      // Liefert n als 12-er-Zahl (als Zahl im Zahlensystem mit
6      // der Basis 12), negative Zahlen mit einem Minuszeichen - davor.
7      // Beispiele:
8      // als12erZahl( 11) ist gleich "B"
9      // als12erZahl(-11) ist gleich "-B"
10     // als12erZahl( 12) ist gleich "10"
11     // als12erZahl(-12) ist gleich "-10"
12     // als12erZahl( 13) ist gleich "11"
13     // als12erZahl( 22) ist gleich "1A"
14     // als12erZahl(130) ist gleich "AA"
15
16     String[] tab = {"0", "1", "2", "3", "4", "5",
17                     "6", "7", "8", "9", "A", "B"};
18
19     if (n < 0) return "-" + als12erZahl(-n);
20     if (n < 12) return tab[n];
21
22     return als12erZahl(n/12) + tab[n%12];
23
24 } // als12erZahl

```

**Lösung 2 (15 Punkte):**

```

1  // Die Klasse ArrayList<K> implementiert die Schnittstelle List<K>.
2  // Die Klasse ArraySet<K> soll die Schnittstelle Set<K> implementieren.
3  // Sie brauchen nur noch die beiden add-Methoden (mit einem bzw. zwei
4  // Parametern) entsprechend zu vervollstaendigen.
5
6  static class ArraySet<K> extends ArrayList<K> implements Set<K> {
7
8      @Override
9      public boolean add(K k) {
10         if (this.contains(k)) return false;
11         return super.add(k);
12     }
13
14     @Override
15     public void add(int index, K k) {
16         if (this.contains(k)) return;
17         super.add(index, k);
18     }
19 } // class ArraySet<K>

```

**Lösung 3 (15 Punkte) :** Welche der .xml-Dateien sind gültig? Und welche sind ungültig? Geben Sie für jede ungültige Datei (möglichst kurz) den Grund ihrer Ungültigkeit an.

Dokument: **ZitatA.xml**

DOMError (error): Attribute "land" is required and must be specified for element type "autor".

istGueltig(ZitatA.xml): false

Dokument: **ZitatB.xml**

istGueltig(ZitatB.xml): true

Dokument: **ZitatC.xml**

DOMError (error): Attribute "zeit" must be declared for element type "zitat".

istGueltig(ZitatC.xml): false

Dokument: **ZitatD.xml**

DOMError (error): Attribute "nr" is required and must be specified for element type "text".

istGueltig(ZitatD.xml): false

Dokument: **ZitatE.xml**

DOMError (error): Attribute value "z1" of type ID must be unique within the document.

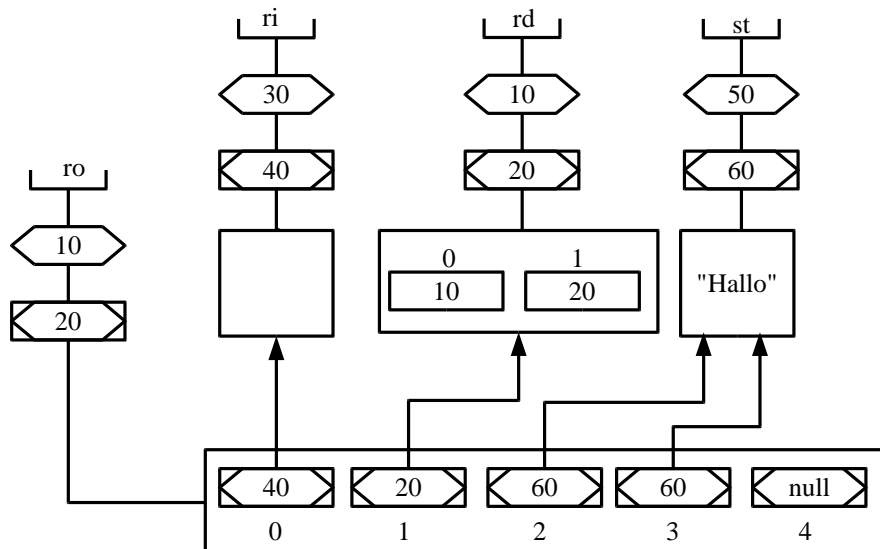
istGueltig(ZitatE.xml): false

**Lösung 4 (20 Punkte):** Stellen Sie die folgenden vier Variablen **rd**, **ri**, **st** und **ro** als Bojen dar (nach Ihrer Wahl in vereinfachter oder in ausführlicher Bojendarstellung):

```

1      double[] rd = {};
2      int[]    ri = {10, 20};
3      String   st = "Hallo";
4
5      Object[] ro = {rd, ri, st, "Hallo", null};

```



**Lösung 5 (15 Punkte): Schreiben Sie eine Methode, die der folgenden Spezifikation entspricht:**

```

1  static public void aufgabe5(Object ob)
2      throws IllegalAccessException, InvocationTargetException
3  {
4      // Wenn ob eine GEEIGNETE add-Funktion enthaelt, dann wird diese
5      // mit den Parametern 3 und 4 aufgerufen und ihr Ergebnis wird
6      // (mit pln) ausgegeben.
7      // Wenn ob keine GEEIGNETE Funktion enthaelt, wird die
8      // Fehlermeldung "Keine add-Methode!" ausgegeben.
9      // GEEIGNET ist eine Funktion, wenn sie folgende Eigenschaften hat:
10     //
11     // 1. Sie hat den Namen add
12     // 2. Sie ist eine oeffentliche Klassenmethode
13     // 3. Sie hat den Rueckgabebetyp int
14     // 4. Sie hat genau 2 Parameter
15     // 5. Diese Parameter sind beide vom Typ int
16     //
17     // Hinweis: Beim Schreiben der Methode brauchen Sie Ausnahmen nicht
18     // zu beruecksichtigen (das erledigt die throws-Klausel bereits).
19
20     Class<?> kob = ob.getClass();
21     Method[] mr = kob.getMethods();
22     for (Method m:mr) {
23         Class<?>[] ptr = m.getParameterTypes();
24         Class<?> ret = m.getReturnType();
25         if (!"add".equals(m.getName())) continue;
26         if (ptr.length != 2) continue;
27         if (ptr[0] != Integer.TYPE) continue;
28         if (ptr[1] != Integer.TYPE) continue;
29         if (ret != Integer.TYPE) continue;
30         int erg = (Integer) m.invoke(null, 3, 4);
31         pln("Ergebnis: " + erg);
32         return;
33     }
34     pln("Keine add-Methode!");
35 } // aufgabe5

```

**Lösung 6 (15 Punkte):** Beantworten Sie die folgenden Fragen möglichst *kurz*, aber *genau* und benutzen Sie dabei die im seminaristischen Unterricht behandelten *Fachbegriffe*:

1. Was ist besser an *Fäden* (engl. threads) als an *Prozessen*?

Ein Fadenwechsel ist viel billiger (d.h. schneller) als ein Prozesswechsel.

2. Angenommen, ein Algorithmus A1 hat eine Zeikomplexität von  $O(n^3)$  und braucht zu Lösung eines Problems der Größe 100 etwa 500 Schritte. Wie viele Schritte braucht A1 dann zur Lösung eines Problems der Größe 200?

4000 Schritte

3. *Wie viele Schritte* braucht man höchstens, um in einer *sortierten Reihung* der Länge 4 Millionen *in einem negativen Fall* zu suchen? Geben Sie die konkrete ("ausgerechnete") Zahl an, keine Formel.

22 Schritte

4. *Welche Tiefe* muss ein binärer Baum, der 8 Millionen Knoten enthält, mindestens haben?

Die Tiefe 23

5. Wenn man einen binären Baum als Reihung `rei` darstellt, welche Indizes haben dann die beiden *Nachfolger* des Knotens `rei[7]`? Und welchen Index hat der *Vorgänger* von `rei[7]`?

Indizes der Nachfolger: 14 und 15. Index des Vorgängers: 3.

6. Welche der folgenden Worte bezeichnen (notwendige oder optionale) Teile eines *Umwandlungsbefehls* im *Formatstring* eines `format`-Befehls?

Länge, Fahrkarte, Schalter, Präzision, Breite, Höhe, Tiefe, Genauigkeit, UZiffer, VZiffer, UBuchstabe, VBuchstabe, Index, Schleife

7. Angenommen, die Variable `kob` zeigt auf ein `Class`-Objekt.

Was liefert dann der Funktionsaufruf `kob.getDeclaredFields()`?

Eine Reihung von `Field`-Objekten, die für jedes in der betreffenden Klasse vereinbarte Attribut (engl. field) ein Objekt enthält. Für geerbte Attribute enthält die Reihungen keine Objekte.

8. *Sortierte Reihungen* werden manchmal als "Ersatz für Sammlungen" verwendet. Nennen Sie zwei *Nachteile*, die sortierte Reihungen in einem solchen Fall haben.

Sie sind aus Beton (ihre Länge ist unveränderbar). Beim Einfügen müssen viele Komponenten verschoben werden (durchschnittlich die Hälfte aller Komponenten). Beim Löschen ebenfalls.