

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 5 und 6 stehen *auf der Rückseite* dieses Blattes!

Aufgabe 1 (20 Punkte): Schreiben Sie eine Methode, die der folgenden Spezifikation entspricht:

```
1  static boolean enthaelt(String[] sra, String[] srb) {
2      // Verlaesst sich darauf, dass sra und srb Mengen von String-Objekten
3      // sind (d.h. keine Doppelgaenger und keine null-Komponenten
4      // enthalten). Liefert true genau dann wenn "sra enthaelt srb" wahr
5      // ist, d.h. wenn es zu jedem String srb[i] einen String sra[j]
6      // gibt, so dass srb[i] und sra[j] gleich sind.
7      // Tip: Unterscheiden Sie zwischen "gleich" und "identisch"!
8      //
9      // Beispiele:
10     // String[] sr0 = {}
11     // String[] sr1 = {"AA", "BB", "CC"};
12     // String[] sr2 = {"CC", "BB"};
13     // String[] sr3 = {"BB", "DD"};
14     // enthaelt(sr1, sr2) ist true
15     // enthaelt(sr1, sr3) ist false
16     // enthaelt(sr3, sr1) ist false
17     // enthalet(sr3, sr0) ist true
18     ...
19 } // enthaelt
```

Aufgabe 2 (20 Punkte): Schreiben Sie eine Methode, die der folgenden Spezifikation entspricht:

```
1  static int valueOf(String s) {
2      // Wandelt s in einen int-Wert um und liefert den als Ergebnis.
3      // Der String s darf nur Dezimalziffern ('0' bis '9') enthalten
4      // und muss mindestens eine Ziffer enthalten. Wenn diese
5      // Bedingungen nicht erfuehlt sind, wird eine Ausnahme des Typs
6      // NumberFormatException geworfen, die die Meldung
7      // "Falscher Param: " + s enthaelt.
8      //
9      // Achtung: Zur Loesung duerfen nur einfache Java-Befehle verwendet
10     // werden und keine Methoden aus der Standardbibliothek, die das
11     // Problem (oder einen grossen Teil davon) loesen (wie z.B.
12     // parseInt, decode, valueOf etc.)
13     ...
14 } // valueOf
```

Aufgabe 3 (15 Punkte): Was gibt die folgende Methode zur Standardausgabe aus?

```

1  static public void aufgabe3() {
2      int n;
3      for (n=0; n<=3; n++) {
4          try {
5              if (n == 1) throw new Error();
6              if (n == 2) throw new Exception();
7              if (n == 3) throw new Throwable();
8          } catch (Error a) {
9              pln("A: n = " + n);
10         } catch (Exception a) {
11             pln("B: n = " + n);
12         } catch (Throwable a) {
13             pln("C: n = " + n);
14             break;
15         } finally {
16             pln("D: n = " + n);
17         } // try/catch
18     } // while
19     pln("E: n = " + n);
20 } // aufgabe3

```

Aufgabe 4 (15 Punkte) Betrachten Sie die folgenden drei Codestücke:

```

1  // -----
2  // Codestueck A:
3  int a1 = 4;
4  int a2 = -3;
5  int a3 = 2;
6  for (int a4=0; a4<=14; a4+=3) {
7      a1 = a1 + a2;
8      a2 = a2 * -1;
9      a3 = a1 + a3 - a4;
10     pln("aus: " + a3);
11 }
12 // -----
13 // Codestueck B:
14 String s = "X1X2X3X4X5X";
15 int n = s.length();
16 while (true) {
17     n = s.lastIndexOf('X', n-1);
18     if (n==-1) break;
19     printf("Ab %d: %s\n", n, s.substring(n));
20 }
21 // -----
22 // Codestueck C:
23 for (int i=5; i>=1; i--) {
24     p(6-i + " ");
25     for (int j=0; j<i; j++) p('O');
26     for (int j=0; j<i; j++) p('X');
27     pln();
28 }
29 // -----

```

Geben Sie für jedes Codestück an, was es (zur Standardausgabe) ausgibt. Schreiben Sie dabei **besonders deutlich**, so dass klar zu erkennen ist, wie viele **Zeilen**, wie viele **Zeichen** und **welche** Zeichen ausgegeben werden (in Zweifelsfällen sollten Sie ihre Lösung noch mal sauber abschreiben).

Aufgabe 5 (15 Punkte): Betrachten Sie die folgende Befehlsfolge:

```
1      String    st = "AB";
2      String[]  sr = {"AB", "CD"};
3      char[]    cr = {};
4      Object[]  or = {null, st, null, sr, cr};
```

Stellen Sie die vier Variablen `st`, `sr`, `cr` und `or` als Bojen dar.

Aufgabe 6: (15 Punkte) Beantworten Sie die folgenden Fragen möglichst *kurz*, aber *genau*. Benutzen Sie dabei die in der Vorlesung eingeführten *Fachbegriffe*.

1. Beschreiben Sie den Vorteil von *Typen* (in einer Programmiersprache wie C oder Java) mit zwei oder drei (nicht zu langen) Sätzen. Diese Sätze sollen für Ihren Manager (der nur oberflächliche Informatikkenntnisse hat und nicht programmieren kann) möglichst gut *verständlich* sein.
2. Angenommen, Sie sehen in einem korrekten Java-Programm (welches auch die üblichen Namenskonventionen einhält) folgenden Befehl:

```
alfred.bertha = Carola.dieter();
```

Was für ein Ding bezeichnet der Name `alfred`? Und `bertha`? Und `Carola`? Und `dieter`?

3. Betrachten Sie die folgende Variablenvereinbarung:

```
int[][][] emil = {{{1,2}, {3,4,5}}, {{6}, {7,8,9}, {10}}};
```

Von welchem *Typ* sind die *Komponenten* von `emil`? Geben Sie den Namen des Typs *zweimal* an: So, wie man ihn z.B. in einer Variablenvereinbarung *schreibt* und so, wie man ihn *ausspricht*.

4. Geben Sie fünf *Literale* an, von jedem der folgenden Typen eines:

`int`, `long`, `double`, `float`, `char`, `String`

5. Wie sieht die Vereinbarung eines *paketweit erreichbaren* Klassenattributs namens `lisa` vom Typ `long` mit dem Anfangswert siebzehn aus?

6. Was kann ein Java-Paket alles enthalten?

7. In Java werden (leider) zwei *Arten von Ausnahmen* unterschieden. Wie bezeichnet man sie?

8. Bei einer Ausführung eines Java-Programms können sogenannten *Ereignisse* (engl. events) eintreten. Was ist die wichtigste, charakteristische *Eigenschaft* solcher Ereignisse?

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 (20 Punkte): Schreiben Sie eine Methode entsprechend der folgenden Spezifikation:

```

1  static boolean enthaelt(String[] sra, String[] srb) {
2      // Verlaesst sich darauf, dass sra und srb Mengen von String-Objekten
3      // sind (d.h. keine Doppelgaenger und keine null-Komponenten
4      // enthalten). Liefert true genau dann wenn "sra enthaelt srb" wahr
5      // ist, d.h. wenn es zu jedem String srb[i] einen String sra[j]
6      // gibt, so dass srb[i] und sra[j] gleich sind.
7      // Tip: Unterscheiden Sie zwischen "gleich" und "identisch"!
8      //
9      // Beispiele:
10     String[] sr0 = {};
11     String[] sr1 = {"AA", "BB", "CC"};
12     String[] sr2 = {"CC", "BB"};
13     String[] sr3 = {"BB", "DD"};
14     // enthaelt(sr1, sr2) ist true
15     // enthaelt(sr1, sr3) ist false
16     // enthaelt(sr3, sr1) ist false
17     // enthalet(sr3, sr0) ist true
18
19     s1: for (String sb : srb) {
20         for (String sa : sra) {
21             if (sb.equals(sa)) continue s1;
22         }
23         return false;
24     }
25     return true;
26 } // enthaelt

```

Lösung 2 (20 Punkte): Schreiben Sie eine Methode, die der folgenden Spezifikation entspricht:

```

1  static int valueOf(String s) {
2      // Wandelt s in einen int-Wert um und liefert den als Ergebnis.
3      // Der String s darf nur Dezimalziffern ('0' bis '9') enthalten
4      // und muss mindestens eine Ziffer enthalten. Wenn diese
5      // Bedingungen nicht erfuehlt sind, wird eine Ausnahme des Typs
6      // NumberFormatException geworfen, die die Meldung
7      // "Falscher Param: " + s enthaelt.
8      //
9      // Achtung: Zur Loesung duerfen nur einfache Java-Befehle verwendet
10     // werden und keine Methoden aus der Standardbibliothek, die das
11     // Problem (oder einen grossen Teil davon) loesen (wie z.B.
12     // parseInt, decode, valueOf etc.)
13
14     NumberFormatException ex =
15         new NumberFormatException("Falscher Param: " + s);
16
17     if (s.length()==0) throw ex;
18
19     int erg = 0;
20
21     for (int i=0; i<s.length(); i++) {
22         char c = s.charAt(i);
23         if (c<'0' || '9'<c) throw ex;
24         c = (char) (c - '0');
25         erg = erg*10 + c;
26     }
27
28     return erg;
29 } // valueOf

```

Lösung 3 (15 Punkte): Was gibt die folgende Methode zur Standardausgabe aus?

```

1  static public void aufgabe3() {
2      int n;
3      for (n=0; n<=3; n++) {
4          try {
5              if (n == 1) throw new Error();
6              if (n == 2) throw new Exception();
7              if (n == 3) throw new Throwable();
8          } catch (Error a) {
9              pln("A: n = " + n);
10         } catch (Exception a) {
11             pln("B: n = " + n);
12         } catch (Throwable a) {
13             pln("C: n = " + n);
14             break;
15         } finally {
16             pln("D: n = " + n);
17         } // try/catch
18     } // while
19     pln("E: n = " + n);
20 } // aufgabe3

```

Ausgabe:

```

D: n = 0
A: n = 1
D: n = 1
B: n = 2
D: n = 2
C: n = 3
D: n = 3
E: n = 3

```

Lösung 4 (15 Punkte) Betrachten Sie die folgenden drei Codestücke:

```

1  // -----
2  // Codestueck A:
3  int a1 = 4;
4  int a2 = -3;
5  int a3 = 2;
6  for (int a4=0; a4<=14; a4+=3) {
7      a1 = a1 + a2;
8      a2 = a2 * -1;
9      a3 = a1 + a3 - a4;
10     pln("aus: " + a3);
11 }
12 // -----
13 // Codestueck B:
14 String s = "X1X2X3X4X5X";
15 int n = s.length();
16 while (true) {
17     n = s.lastIndexOf('X', n-1);
18     if (n==-1) break;
19     printf("Ab %d: %s%n", n, s.substring(n));
20 }
21 // -----
22 // Codestueck C:
23 for (int i=5; i>=1; i--) {
24     p(6-i + " ");
25     for (int j=0; j<i; j++) p('O');
26     for (int j=0; j<i; j++) p('X');
27     pln();
28 }
29 // -----

```

Geben Sie für jedes Codestück an, was es (zur Standardausgabe) ausgibt. Schreiben Sie dabei **besonders deutlich**, so dass klar zu erkennen ist, wie viele **Zeilen**, wie viele **Zeichen** und **welche** Zeichen ausgegeben werden (in Zweifelsfällen sollten Sie ihre Lösung noch mal sauber abschreiben).

Ausgaben:

Codestueck A:

```
aus: 3
aus: 4
aus: -1
aus: -6
aus: -17
```

Codestueck B:

```
Ab 10: X
Ab 8: X5X
Ab 6: X4X5X
Ab 4: X3X4X5X
Ab 2: X2X3X4X5X
Ab 0: X1X2X3X4X5X
```

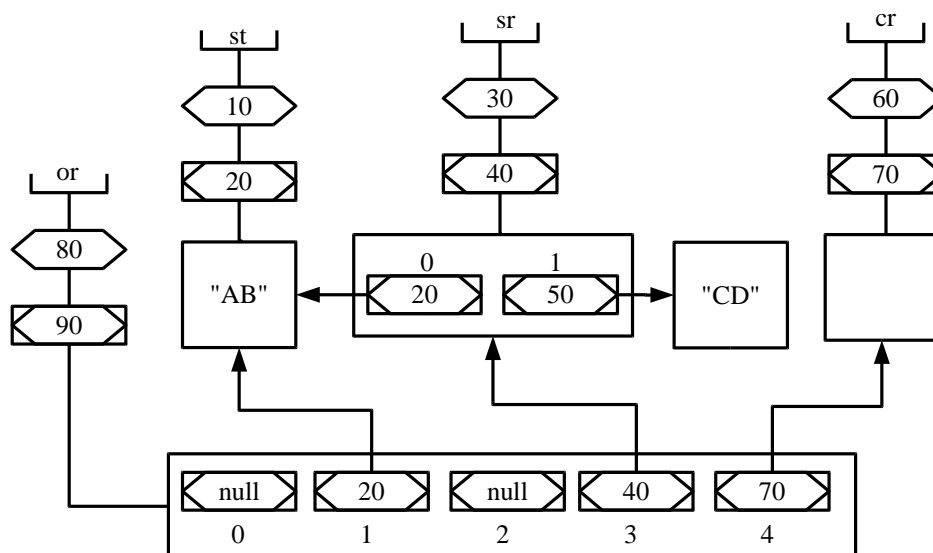
Codestueck C:

```
1 OOOOXXXXX
2 OOOOXXXX
3 OOOXXX
4 OOX
5 OX
```

Lösung 5 (15 Punkte): Betrachten Sie die folgende Befehlsfolge:

```
1 String st = "AB";
2 String[] sr = {"AB", "CD"};
3 char[] cr = {};
4 Object[] or = {null, st, null, sr, cr};
```

Stellen Sie die vier Variablen st, sr, cr und or als Bojen dar.



Lösung 6: (15 Punkte) Beantworten Sie die folgenden Fragen möglichst kurz, aber genau. Benutzen Sie dabei die in der Vorlesung eingeführten Fachbegriffe.

1. Beschreiben Sie den Vorteil von *Typen* (in einer Programmiersprache wie C oder Java) mit zwei oder drei (nicht zu langen) Sätzen. Diese Sätze sollen für Ihren Manager (der nur oberflächliche Informatikkenntnisse hat und nicht programmieren kann) möglichst gut verständlich sein.

Typen machen es möglich, dass bestimmte Fehler des Programmierers automatisch erkannt werden. Dadurch wird das Beseitigen dieser Fehler billiger.

2. Angenommen, Sie sehen in einem korrekten Java-Programm (welches auch die üblichen Namenskonventionen einhält) folgenden Befehl:

```
alfred.bertha = Carola.dieter();
```

Was für ein Ding bezeichnet der Name *alfred*? Und *bertha*? Und *Carola*? Und *dieter*?

alfred bezeichnet ein Objekt (einen Modul)

bertha bezeichnet ein Attribut (engl. field) des Objekts *alfred*

Carola bezeichnet eine Klasse (einen Modul)

dieter bezeichnet eine Klassenmethode

3. Betrachten Sie die folgende Variablenvereinbarung:

```
int[][][] emil = {{{1,2}, {3,4,5}}, {{6}, {7,8,9}, {10}}};
```

Von welchem Typ sind die Komponenten von *emil*? Geben Sie den Namen des Typs *zweimal* an: So, wie man ihn z.B. in einer Variablenvereinbarung *schreibt* und so, wie man ihn *ausspricht*.

Geschrieben: int[][] Gesprochen: Reihung von Reihungen von int (-Variablen)

4. Geben Sie fünf *Literale* an, von jedem der folgenden Typen eines:

int, *long*, *double*, *float*, *char*, *String*

17, 25L, 12.345, 123.45F, 'X', "ABC"

5. Wie sieht die Vereinbarung eines paketweit erreichbaren Klassenattributs namens *lisa* vom Typ *long* mit dem Anfangswert siebzehn aus?

static long *lisa* = 17L;

6. Was kann ein Java-Paket alles enthalten?

Klassen, Schnittstellen und Pakete.

7. In Java werden (leider) zwei *Arten von Ausnahmen* unterschieden. Wie bezeichnet man sie?

Geprüfte und ungeprüfte Ausnahmen (engl. checked and unchecked exceptions)

8. Bei einer Ausführung eines Java-Programms können sogenannten *Ereignisse* (engl. events) eintreten. Was ist die wichtigste, charakteristische *Eigenschaft* solcher Ereignisse?

Ereignisse sind nicht wiederholbar (sie sind "historisch").