

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*). Mehrere Aufgaben (-teile) stehen *auf der Rückseite* dieses Blattes!**Aufgabe 1 (20 Punkte):** Schreiben Sie eine Methode entsprechend der folgenden Spezifikation:

```

1  static public int quersumme(int zahl, short B) {
2  // Welche Quersumme hat die zahl als B-er-Zahl (d.h. wenn man sie
3  // im Zahlensystem mit der Basis B darstellt)? Diese Funktion liefert
4  // die Antwort.
5  // Falls die zahl negativ ist, wird die Quersumme von -zahl als
6  // Ergebnis geliefert.
7  // Falls die Basis B nicht im Bereich 2 bis Short.MAX_VALUE liegt,
8  // wird -1 als Ergebnis geliefert.
9
10 // Beispiele:
11 // short b01 = 1;           // Eine zu kleine Basis
12 // short b02 = 2;           // Die kleinste zulaessige Basis
13 // short b05 = 5;
14 // short b10 = 10;
15 // short b16 = 16;
16 // short bmax = Short.MAX_VALUE; // Die groesste zulaessige Basis
17 //
18 // quersumme(16 ,      b02 ) ist gleich  1
19 // quersumme(16 ,      b05 ) ist gleich  4
20 // quersumme(16 ,      b10 ) ist gleich  7
21 // quersumme(16 ,      b16 ) ist gleich  1
22 // quersumme(16 ,      bmax) ist gleich 16
23 //
24 // quersumme(b02 ,      b02 ) ist gleich  1
25 // quersumme(b05 ,      b05 ) ist gleich  1
26 // quersumme(b10 ,      b10 ) ist gleich  1
27 // quersumme(b16 ,      b16 ) ist gleich  1
28 // quersumme(bmax ,     bmax) ist gleich  1
29 //
30 // quersumme(255 ,      b02 ) ist gleich  8
31 // quersumme( 86 ,      b05 ) ist gleich  6
32 // quersumme(123 ,      b10 ) ist gleich  6
33 // quersumme(170 ,      b16 ) ist gleich 20
34 // quersumme(123 ,      b01 ) ist gleich -1
35 // ...
36 } // quersumme

```

Aufgabe 2 (20 Punkte) Schreiben Sie zwei Methoden entsprechend den folgenden Spezifikationen:

```

1  static public boolean istGrossbuchstabe(char zeichen) {
2  // Liefert true genau dann wenn zeichen ein Grossbuchstabe ist.
3  // Als Grossbuchstaben gelten hier die Zeichen 'A', 'B', ..., 'Z'.
4  // Zusammengesetzte Anweisungen sind in dieser Methode nicht erlaubt!
5  // ...
6  } // istGrossbuchstabe
7

```

```

8  static public int anzahlGrossbuchstaben(ArrayList<String> als) {
9  // Liefert die Anzahl der Grossbuchstaben ('A', 'B', ..., 'Z')
10 // die in den Komponenten von als enthalten sind.
11 // Beispiel:
12 // ArrayList<String> a01 = new ArrayList<String>();
13 // als.add("Hallo Harry!");
14 // als.add("Wie geht's?");
15 // als.add("Danke, gut!");
16 //
17 // Fuer die Sammlung a01 gilt jetzt:
18 // anzahlGrossbuchstaben(a01) ist gleich 4.
19 // ...
20 } // anzahlGrossbuchstaben

```

Aufgabe 3 (15 Punkte): Stellen Sie die folgende (zweistufige) Reihung r2s als Boje dar und beantworten Sie die vier unten stehenden Fragen:

```

1  String[][] r2s = {
2  {"AB", null, "CD"},
3  null,
4  {null},
5  {}
6  };

```

Frage 3.1: Welche Länge hat die Reihung r2s?

Frage 3.2: Welche Länge hat die Reihung r2s[0]?

Frage 3.3: Welche Länge hat die Reihung r2s[2]?

Frage 3.4: Welche Länge hat die Reihung r2s[3]?

Aufgabe 4 (15 Punkte): Betrachten Sie die folgenden Klassenvereinbarungen:

```

1  class K1 {
2  static private int sum1 = 0;
3  static private int sum2 = 0;
4  private int zal1;
5  private int zal2;
6
7  public K1(int zal1, int zal2) {
8  this.zal1 = zal1;
9  this.zal2 = zal2;
10 sum1 += zal1;
11 sum2 += zal2;
12 }
13
14 public K1() {}
15 }
16
17 class K2 {
18 static public void main(String[] _) {
19 K1 ob1 = new K1(10, 20);
20 K1 ob2 = new K1();
21 K1 ob3 = new K1();
22
23 K1 ob4 = new K1(30, 40);
24 K1 ob5 = new K1();
25
26 } // main
27 }

```

4.1. Wie viele *Klassenelemente* werden in der Klasse K1 vereinbart?4.2. Wie viele *Elemente* werden in jedes K1-Objekt eingebaut?4.3. Wie viele *Konstruktoren* werden in der Klasse K1 vereinbart?

4.4. Sei M1 der Moment, in dem der Ausführer die Zeile 21 fertig ausgeführt hat. Wie viele Module existieren in diesem Moment M1 und wie heißen diese Module?

4.5. Wie viele `int`-Variablen existieren im Moment M1 und wie heißen diese Variablen mit vollen Namen?

4.6. Sei M2 der Moment, in dem der Ausführer die Zeile 24 fertig ausgeführt hat. Wie viele Module existieren in diesem Moment M2 und wie heißen diese Module?

4.7. Wie viele `int`-Variablen existieren im Moment M2 und wie heißen diese Variablen mit vollen Namen?

Aufgabe 5 (15 Punkte): Geben Sie von jeder der folgenden 3 Schleifen an, was sie zum Bildschirm ausgibt.

```

1   final int z = 3;
2
3   // Schleife 5.1.:
4   for (int i=1; i<=z; i++) {
5       for (int j=i; j>=1; j--) p("XX");
6       pln();
7   }
8
9   // Schleife 5.2:
10  for (int i=0; i<2*z; i++) {
11      for (int j=i%z+1; j>=1; j--) p("XX");
12      pln();
13  }
14
15  // Schleife 5.3:
16  for (int i=1; i<=z; i++) {
17      for (int j=i-1; j>=1; j--) p("OO");
18      pln("XX");
19  }
20
21  // Schleife 5.4:
22  for (int i=1; i<=z; i++) {
23      for (int j=z-i; j>=1; j--) p("OO");
24      for (int j=1; j<=i+(i-1); j++) p("XX");
25      for (int j=z-i; j>=1; j--) p("OO");
26      pln();
27  }

```

Die Namen `p` und `pln` sind auch hier Abkürzungen für die Namen `System.out.print` und `System.out.println`.

Aufgabe 6: (15 Punkte)

6.1. Was für Elemente darf der Programmierer innerhalb einer *Java-Schnittstelle* (engl. interface) vereinbaren? Geben Sie die *Erreichbarkeit*, die *Aspektzugehörigkeit* und die *Art* der Elemente an.

6.2. Was ist ein *Behälterobjekt* (engl. container object)?

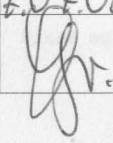
6.3. Was ist die wichtigste Eigenschaft eines *Ereignisses* (engl. of an event)?

6.4 Was wissen Sie über die Ströme eines Typs, dessen Name mit `Reader` endet (z.B. `FileReader` oder `LineNumberReader`)?

6.5. Welche Ausnahmen (eigentlich: Ausnahmeklassen) *muss man* in der `throws`-Klausel am Anfang einer Methode `m` angeben?

6.6. Welche Klassen gehören (als Hauptklasse bzw. als Nebenklassen) zu einem Programm namens `Carola`?

Korrekturergebnis:

A1	19	
A2	19	
A3	15	
A4	12	
A5	11	
A6	11	
Summe	87	
Note	1,7	
Datum	17.07.08	
Unterschrift		

① static public int quersumme (int zahl, short B) {

int erg = 0;

int ziff = 0;

if (zahl < 0) zahl = zahl * (-1); zahl = -zahl;

if ((B < 2) || (B > Short.MAX_VALUE)) return -1;

while (true) {

ziff = zahl % B;

zahl = zahl / B;

erg += ziff;

if (zahl == 0) break;

}

return erg;

}

19

```

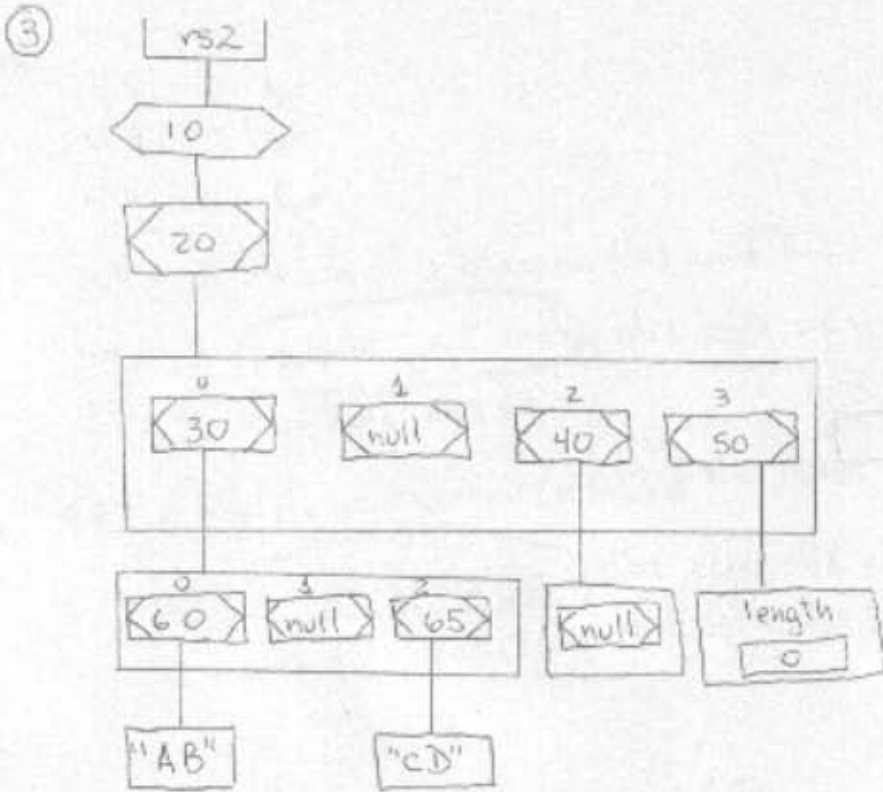
(2) static public boolean istGrossbuchstabe(char zeichen) {
    return ((zeichen >= 'A') && (zeichen <= 'Z'));
}

```

```

static public int anzahlGrossbuchstaben(ArrayList<String> als) {
    int anzahl = 0;
    for (String elem: als) {
        for (i=0; i<elem.length(); i++) {
            if (istGrossbuchstabe(elem.charAt(i)) = true) anzahl++;
            -1
        }
    }
    ret anzahl;
}

```



- 3.1 - Länge 4 ✓
- 3.2 - Länge 3 ✓
- 3.3 - Länge 2 ✓
- 3.4 - Länge 0 ✓

15

- ④
- 4.1 - 2 ✓
 - 4.2 - 2 ✓
 - 4.3 - 2 ✓
 - 4.4 - 5 Module: 2 Klassen (K1 und K2) und 3 Objekte (ob1, ob2, ob3); ✓
 - 4.5 - 8 K1.zal1, K1.zal2, ob1.zal1, ob1.zal2, ob2.zal1, ob2.zal2, -3
ob3.zal1, ob3.zal2
 - 4.6 - 7 Module: 2 Klassen (K1 und K2) und 5 Objekte (ob1, ob2, ob3, ob4, ob5) ✓
 - 4.7 - 12 K1.zal1, K1.zal2 ob1.zal1, ob1.zal2, ob2.zal1, ob2.zal2, ob3.zal1,
ob3.zal2, ob4.zal1, ob4.zal2, ob5.zal1, ob5.zal2

⑤ Schleife 5.1:

XX
XXXX
XXXXXX



Schleife 5.3:

XX
O O X X
O O O O X X



Schleife 5.2

XX
XXXX
XXXXXXXX
XX

f - 4

Schleife 5.4:

O O O O X X O O O O
O O X X X X X O O
X X X X X X X X X



AA

⑥

6.1- Öffentliche Objektmethoden
Öffentliche Klassenkonstanten ✓

6.2- Ist ein Ding die andere Dinge enthält: z.B: Variablen, Bildschirm, usw.
f - 2

6.3- Ein Ereigniss findet an einem bestimmten Ort und zu einer bestimmten Zeit statt und kann nicht wiederholt werden. ✓

6.4- Es handelt sich um zeichenorientierte Eingabeströme. ✓

6.5 Alle (Klassen von) geprüften Ausnahmen, die in der Methode m möglicherweise geworfen, aber dort nicht gefangen werden ✓

6.6- Die Hauptklasse ist die, die eine main-Methode enthält, alle anderen sind Nebenklassen - 2

welche?

11