

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 4 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("00");
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("00");
24         for (int j=1; j<=i+(i-1); j++) p("XX");
25         for (int j=Z-i; j>=1; j--) p("00");
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`?

Lösung 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     if (B < 2) return -1;
11
12     int erg = 0;
13     int ziff;
14     zahl=Math.abs(zahl);
15
16     while (true) {
17         if (zahl == 0) break;
18         ziff = zahl % B;
19         zahl = zahl / B;
20         erg += ziff;
21     }
22
23     return erg;
24 } // quersumme
```

Lösung 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
5      return 'A' <= zeichen && zeichen <= 'Z';
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     int erg = 0;
12     for (String s: als) {
13         for (int i=0; i<s.length(); i++) {
14             if (istGrossbuchstabe(s.charAt(i))) erg++;
15         }
16     }
17     return erg;
18 } // anzahlGrossbuchstaben
```

Lösung 3 (15 Punkte): Stellen Sie die folgende Reihung r2s als Boje dar:

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

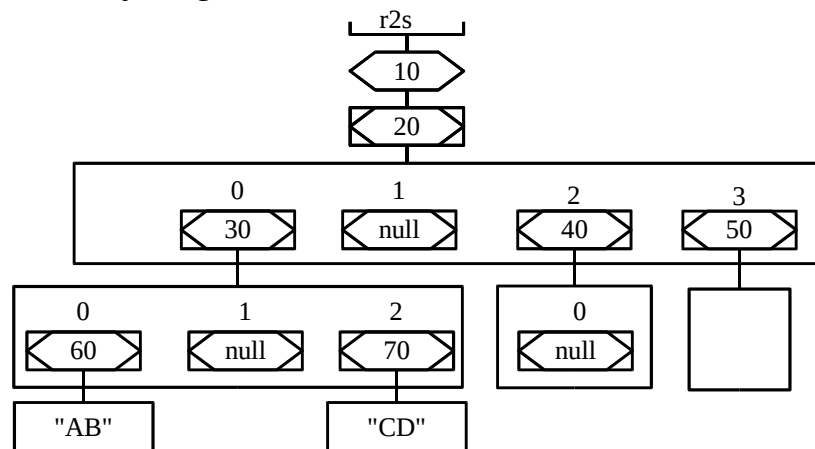
Frage 3.1: Welche Länge hat die Reihung r2s? 4

Frage 3.2: Welche Länge hat die Reihung r2s[0]? 3

Frage 3.3: Welche Länge hat die Reihung r2s[2]? 1

Frage 3.4: Welche Länge hat die Reihung r2s[3]? 0

Die Reihung r2s als Boje dargestellt:



Lösung 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? 2

4.2. Wie viele *Elemente* werden in jedes K1-Objekt eingebaut? 2

4.3. Wie viele *Konstruktoren* werden in der Klasse K1 vereinbart? 2

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?

5 Module, 2 Klassen (K1 und K2) und 3 Objekte (ob1, ob2, ob3).

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

8 `int`-Variablen: K1.sum1, K1.sum2, ob1.zal1, ob1.zahl2, ob2.zal1, ob2.zal2, ob3.zal1, ob3.zal2.

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?

7 Module, zusätzlich zu den 5 Modulen aus Lösung 4.4.: ob4, ob5.

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

12 `int`-Variablen, zusätzlich zu den 8 Variablen aus Lösung 4.5.:
ob4.zal1, ob4.zahl2, ob5.zal1, ob5.zal2.

Lösung 5 (15 Punkte): Geben Sie von jeder der folgenden 4 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("00");
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("00");
24         for (int j=1; j<=i+(i-1); j++) p("XX");
25         for (int j=Z-i; j>=1; j--) p("00");
26         pln();
27     }

```

Ausgabe der Schleife 5.1.:

XX
XXXX
XXXXXX

Ausgabe der Schleife 5.2.:

XX
XXXX
XXXXXX
XX
XXXX
XXXXXX

Ausgabe der Schleife 5.3.:

XX
00XX
0000XX

Ausgabe der Schleife 5.4.:

0000XX0000
00XXXXXXXX00
XXXXXXXXXXXX

Lösung 6: (15 Punkte)

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

Öffentliche (abstrakte) Objektmethoden (public object methods)

Öffentliche (unveränderbare) Klassenattribute (public class fields)

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

Ein Behälter ist ein Grabo-Objekt, in welches man andere Grabo-Objekte hineintun kann.

6.3. Was ist die *wichtigste Eigenschaft* eines Ereignisses?

Ein Ereignis findet an einem bestimmten Ort und zu einer bestimmten Zeit statt und kann nicht wiederholt werden.

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

Es handelt sich um zeichenorientierte Eingabeströme.

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

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

6.6. Welche Klassen gehören zu einem Programm namens `Carola`?

Eine Hauptklasse namens `Carola` (mit einer `main`-Methode darin) und (als Nebenklassen) alle Klassen die benötigt werden, um die Methode `Carola.main` auszuführen.