

Vorname (bitte deutlich und lesbar)

Nachname (bitte deutlich und lesbar)

Matrikel-Nr (bitte deutlich und lesbar)

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*). Diese Klausur besteht aus 6 Aufgaben.

Aufgabe 1 (20 Punkte): Schreiben Sie eine Methode entsprechend der folgenden Spezifikation:

```

1  static public int anzAB(ArrayList<char[][]> al) {
2      // Wie oft kommen die Zeichen 'A' und 'B' in al unmittelbar nachein-
3      // ander vor? Diese Funktion liefert die Antwort.
4      //
5      // Beispiel:
6      // char[][] crr1 = {{'X', 'A', 'B', 'X', 'A', 'B'}, {'A', 'B', 'C'}};
7      // char[][] crr2 = {{'A', 'A', 'B'}, {'B', 'A', 'B'}, {'X', 'Y'}};
8      // ArrayList<char[][]> alf = new ArrayList<char[][]>();
9      // alf.add(crr1);
10     // alf.add(crr2);
11     //
12     // Jetzt hat der Ausdruck anzAB(alf) den Wert 5.
13     ...
14 } // anzAB

```

Aufgabe 2 (20 Punkte) Schreiben Sie eine Methode entsprechend der folgenden Spezifikation:

```

1  static int[] histoZahlen(char[] cr) {
2      // Die Komponenten von cr sollte man hier einfach als Zahlen
3      // verstehen, nicht als Zeichen.
4      //
5      // Sei max die groesste Zahl in cr. Dann liefert diese Funktion eine
6      // Reihung erg der Laenge max+1 und es gilt:
7      // In erg[0] steht, wie haeufig die Zahl 0 in cr vorkommt.
8      // In erg[1] steht, wie haeufig die Zahl 1 in cr vorkommt.
9      // ...
10     // In erg[max] steht, wie haeufig die Zahl max in cr vorkommt.
11     //
12     // Beispiel:
13     // char[] par1 = {2, 2, 0, 3, 0, 3, 2, 2, 2};
14     // int [] erg1 = histoZahlen(par1);
15     // Dann ist max gleich 3 und erg1 ist gleich {2, 0, 5, 2}
16     ...
17 } // histoZiffern

```

Aufgabe 3 (15 Punkte): Betrachten Sie das folgende Programm:

```

1  static class K77 {
2      public String s1 = "ABC";
3      public String s2 = "ABC";
4      public String s3 = new String("ABC");
5
6      static public void main(String[] _) {
7          K77 ob77 = new K77();
8          String tmp = ob77.s2;
9          ob77.s2 = ob77.s3;
10         ob77.s3 = tmp;
11     }
12 }

```

3.1. Wie sieht die Variable ob77 (als Boje dargestellt) aus, wenn die Zeile 7 fertig ausgeführt ist?

3.2. Wie sieht die Variable ob77 (als Boje dargestellt) aus, wenn die Zeile 10 fertig ausgeführt ist?

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

```

1  static class K1 {}
2  static class K2 extends K1 {}
3  static class K3 extends K1 {}
4  static class K4 extends K3 {}
5
6  static K1 v1 = new K1();
7  static K2 v2 = new K2();
8  static K3 v3 = new K3();
9  static K4 v4 = new K4();
10
11 static K1 v12 = v2;
12 static K1 v13 = v3;
13 static K1 v14 = v4;
14
15 static K2 v22 = v2;
16
17 static K3 v33 = v3;

```

Füllen Sie die folgende Tabelle (mit "JA" und "NEIN") aus:

Befehl	Zur Compile-Zeit		Zur Laufzeit
	Ist die Zuweisung erlaubt?	Ist der Cast erlaubt?	Funktioniert der Cast?
<code>v2 = (K2) v12;</code>			
<code>v2 = (K1) v12;</code>			
<code>v3 = (K3) v22;</code>			
<code>v4 = (K4) v33;</code>			
<code>v2 = (K3) v22;</code>			
<code>v2 = (K2) v14;</code>			

Aufgabe 5 (15 Punkte): Was geben die folgenden 4 Schleifen zum Bildschirm aus?

```

1  // Schleife 5.1:
2  for (int i=-3; i<5; i+=3) {
3      p(i + 2 + " ");
4      if (i%2 == 0) i-=2;
5  }
6  pln();
7
8  // Schleife 5.2:
9  int[] rei1 = {10, 20, 30};
10 for (int n1 : rei1) {
11     for (int n2 : rei1) {
12         p(" " + n1 + n2 + " ");
13     }
14     pln();
15 }

```

// Fortsetzung auf der Rückseite!

```
16
17 // Schleife 5.3:
18 long[] rei2 = {30, 20, 10};
19 for (long n : rei2) {
20     n++;
21     p(n + " ");
22 }
23 pLn();
24 for (long n : rei2) {
25     p(n + " ");
26 }
27 pLn();
28
29 // Schleife 5.4:
30 StringBuilder sb = new StringBuilder("ABCDE");
31 for (int i=0; i<sb.length(); i++) {
32     p(sb.charAt(i) + "+");
33     sb.deleteCharAt(i);
34 }
35 pLn();
```

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. Betrachten Sie das folgende Programm:

```
1 class K1 {
2     K2 obA = new K2();
3     K2 obB = obA;
4     K2 obC;
5
6     static public void main(String[] _) {
7         K2 obD = new K2();
8         K1 obE = new K1();
9         ...
10    }
11 }
12
13 class K2 {
14     int n = 17;
15     char c = 'A';
16 }
```

6.1.1. Wie viele Module existieren, wenn der Ausführer die Zeile 7 fertig ausgeführt hat und wie heißen diese Module?

6.1.2. Ebenso für Zeil 8.

6.2. Was ist eine Sammlung?

6.3. Übersetzen Sie die folgenden beiden Befehle ins Deutsche:

6.3.1. `double dora = 2.0 * Math.PI;`

6.3.2. `dora = (dora + 2.5) * 3.0;`

6.4. Warum gibt es zur Schnittstelle `MouseWheelListener` (oder: zur Ereignis-Oberart *Mausrad-Ereignis*) keine Adapterklasse?

6.5. Wenn wir die Elemente einer Klasse nach ihrer Art in Gruppen einteilen, wie viele und welche Gruppen ergeben sich dann?

6.6. Wann wird eine Nebenklasse eines Java-Programms geladen?

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 public int anzAB(ArrayList<char[][]> al) {
2      // Wie oft kommen die Zeichen 'A' und 'B' in al unmittelbar nachein-
3      // ander vor?. Diese Funktion liefert die Antwort.
4      //
5      // Beispiel:
6      // char[][] crr1 = {{'X', 'A', 'B', 'X', 'A', 'B'}, {'A', 'B', 'C'}};
7      // char[][] crr2 = {{'A', 'A', 'B'}, {'B', 'A', 'B'}, {'X', 'Y'}};
8      // ArrayList<char[][]> alf = new ArrayList<char[][]>();
9      // alf.add(crr1);
10     // alf.add(crr2);
11     //
12     // Jetzt hat der Ausdruck anzAB(alf) den Wert 5.
13
14     int anz = 0;
15
16     for (char[][] crr : al) {
17         for (char[] cr : crr) {
18             for (int i=0; i<cr.length-1; i++) {
19                 if (cr[i] == 'A' && cr[i+1] == 'B') anz++;
20             }
21         }
22     }
23     return anz;
24 } // anzAB

```

Lösung 2 (20 Punkte) Schreiben Sie eine Methode entsprechend den folgenden Spezifikationen:

```

1  static int[] histoZahlen(char[] cr) {
2      // Die Komponenten von cr sollte man hier einfach als Zahlen
3      // verstehen, nicht als Zeichen.
4      //
5      // Sei max die groesste Zahl in cr. Dann liefert diese Funktion eine
6      // Reihung erg der Laenge max+1 und es gilt:
7      // In erg[0] steht, wie haeufig die Zahl 0 in cr vorkommt.
8      // In erg[1] steht, wie haeufig die Zahl 1 in cr vorkommt.
9      // ...
10     // In erg[max] steht, wie haeufig die Zahl max in cr vorkommt.
11
12     int max = -1;
13     for (char c : cr) if (max < c) max = c;
14
15     int[] erg = new int[max+1];
16     for (char c : cr) erg[c]++;
17     return erg;
18 } // histoZiffern

```

Lösung 3 (15 Punkte): Betrachten Sie das folgende Programm:

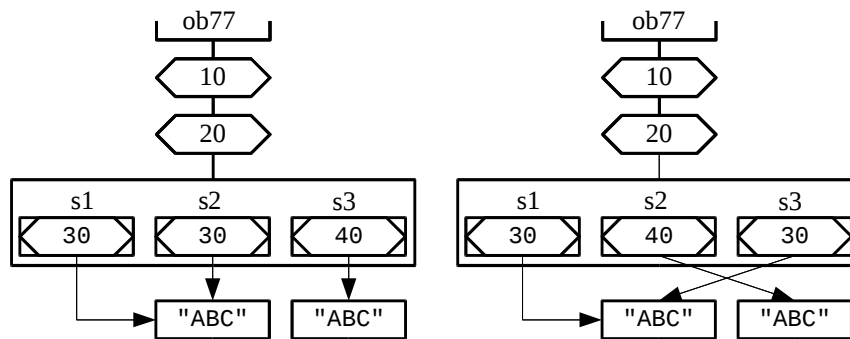
```

1  static class K77 {
2      public String s1 = "ABC";
3      public String s2 = "ABC";
4      public String s3 = new String("ABC");
5
6      static public void main(String[] _) {
7          K77 ob77 = new K77();
8          String tmp = ob77.s2;
9          ob77.s2 = ob77.s3;
10         ob77.s3 = tmp;
11     }
12 }

```

3.1. Wie sieht die Variable ob77 (als Boje dargestellt) aus, wenn die Zeile 7 fertig ausgeführt ist?

3.2. Wie sieht die Variable ob77 (als Boje dargestellt) aus, wenn die Zeile 10 fertig ausgeführt ist?



Lösung 4 (15 Punkte): Betrachten Sie die folgenden Vereinbarungen:

```

1  static class K1 {}
2  static class K2 extends K1 {}
3  static class K3 extends K1 {}
4  static class K4 extends K3 {}
5
6  static K1 v1 = new K1();
7  static K2 v2 = new K2();
8  static K3 v3 = new K3();
9  static K4 v4 = new K4();
10
11 static K1 v12 = v2;
12 static K1 v13 = v3;
13 static K1 v14 = v4;
14
15 static K2 v22 = v2;
16
17 static K3 v33 = v3;

```

Füllen Sie die folgende Tabelle (mit "JA" und "NEIN") aus:

Befehl	Zur Compile-Zeit		Zur Laufzeit
	Ist die Zuweisung erlaubt?	Ist der Cast erlaubt?	Funktioniert der Cast?
<code>v2 = (K2) v12;</code>	Ja	Ja	Ja
<code>v2 = (K1) v12;</code>	Nein	Ja	--
<code>v3 = (K3) v22;</code>	Ja	Nein	--
<code>v4 = (K4) v33;</code>	Ja	Ja	Nein
<code>v2 = (K3) v22;</code>	Nein	Nein	--
<code>v2 = (K2) v14;</code>	Ja	Ja	Nein

Lösung 5 (15 Punkte): Was geben die folgenden 4 Schleifen zum Bildschirm aus?

```

1 // Schleife 5.1:
2 for (int i=-3; i<5; i+=3) {
3     p(i + 2 + " ");
4     if (i%2 == 0) i-=2;
5 }
6 println();
7
8 // Schleife 5.2:
9 int[] rei1 = {10, 20, 30};
10 for (int n1 : rei1) {
11     for (int n2 : rei1) {
12         p("" + n1 + n2 + " ");
13     }
14     println();
15 }
16
17 // Schleife 5.3:
18 long[] rei2 = {30, 20, 10};
19 for (long n : rei2) {
20     n++;
21     p(n + " ");
22 }
23 println();
24 for (long n : rei2) {
25     p(n + " ");
26 }
27 println();
28
29 // Schleife 5.4:
30 StringBuilder sb = new StringBuilder("ABCDE");
31 for (int i=0; i<sb.length(); i++) {
32     p(sb.charAt(i) + "+");
33     sb.deleteCharAt(i);
34 }
35 println();

```

Die Ausgaben der Schleifen:

```
// Schleife 5.1:      -1 2 3 6
```

```
// Schleife 5.2:      1010 1020 1030
                     2010 2020 2030
                     3010 3020 3030
```

```
// Schleife 5.3:      31 21 11
                     30 20 10
```

```
// Schleife 5.4: A+C+E+
```

Lösung 6: (15 Punkte)

6.1. Betrachten Sie das folgende Programm:

```
1 class K1 {
2     K2 obA = new K2();
3     K2 obB = obA;
4     K2 obC;
5
6     static public void main(String[] _) {
7         K2 obD = new K2();
8         K1 obE = new K1();
9         ...
10    }
11 }
12
13 class K2 {
14     int n = 17;
15     char c = 'A';
16 }
```

6.1.1. Wie viele Module existieren, wenn der Ausführer die Zeile 7 fertig ausgeführt hat und wie heißen diese Module?

3 Module: Die Klassen K1 und K2 und das Objekt obD

6.1.2. Ebenso für Zeil 8.

5 Module: Die Klassen K1 und K2 und die Objekte obD, obA und obE.

6.2. Was ist eine Sammlung?

Ein Objekt, in das man andere Objekte hineintun kann, in dem man nach einem Objekt suchen und aus dem man Objekte wieder entfernen kann (etwa so wie Briefmarken bei einer Briefmarkensammlung).

6.3. Übersetzen Sie die folgenden beiden Befehle ins Deutsche:

6.3.1. `double dora = 2.0 * Math.PI;`

Erzeuge eine Variable namens dora vom Typ double und initialisiere sie mit dem Wert des Ausdrucks `2.0 * Math.PI`.

6.3.2. `dora = (dora + 2.5) * 3.0;`

Berechne den Wert des Ausdrucks `(dora + 2.5) * 3.0` und weise ihn der Variablen dora zu.

6.4. Warum gibt es zur Schnittstelle `MouseListener` (oder: zur Ereignis-Oberart *MouseListener*) keine Adapterklasse?

Weil die Schnittstelle `MouseListener` nur eine Methode enthält.

6.5. Wenn wir die Elemente einer Klasse nach ihrer Art in Gruppen einteilen, wie viele und welche Gruppen ergeben sich dann?

4 Gruppen: Attribute, Methoden, Klassen, Schnittstellen.

6.6. Wann wird eine Nebenklasse eines Java-Programms geladen?

Wenn sie zum ersten Mal gebraucht wird.