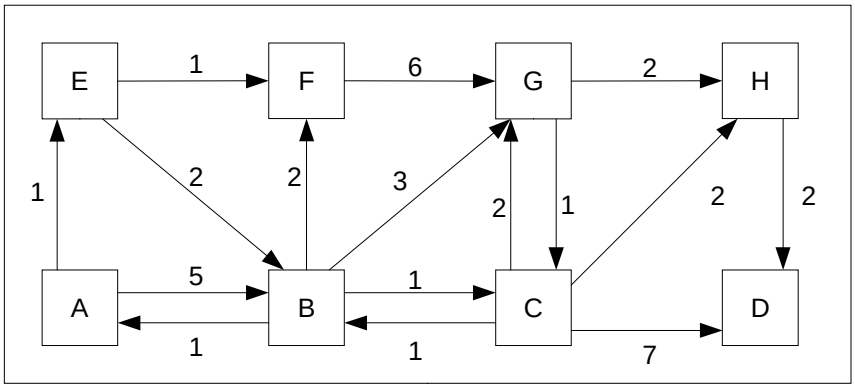
Vorname Nachname Matrikel-Nr

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

Ihre Lösung für Aufgabe 1 können Sie direkt auf dieses bedruckte Blatt schreiben. Schreiben Sie die Lösung für jede andere Aufgabe auf die *Vorderseite* eines Ihrer Lösungsblätter (und lassen Sie die Rückseiten Ihrer Lösungsblätter **leer**).

Aufgabe 1 (20 Punkte): Betrachten Sie den folgenden Graphen:



Teilaufgabe 1.1. Berechnen Sie nach dem Dijkstra-Algorithmus die kürzesten Wege vom Knoten **A** zu allen anderen Knoten, indem Sie die folgende Tabelle ausfüllen (wie im SU und in den Üs behandelt):

besucht	**Knoten**										**Min**
	A										
	B										
	C										
	D										
	E										
	F										
	G										
	H										

Teilaufgabe 1.2.
Über welche Knoten führt der kürzeste Weg von **A** nach **D**?

Aufgabe 2 (20 Punkte): Schreiben Sie eine **rekursive** Methode ("Schleifen dürfen hier nicht rein!") entsprechend der folgenden Spezifikation:

```
1      static int anz5bis9(final int N) {
2         // Verlaesst sich darauf, dass N groesser oder gleich 0 ist.
3         // Liefert die Antwort auf folgende Frage: Wenn man N als
4         // 10-er-Zahl (Dezimalzahl) darstellt, wie viele Ziffern sind
5         // dann groesser als 4?
6         //
7         // Beispiele:
8         // anz5bis9(234) ist gleich 0
9         // anz5bis9(352) ist gleich 1
10        // anz5bis9(960) ist gleich 2
11        // anz5bis9(585) ist gleich 3
12        // anz5bis9(  1) ist gleich 0
13        // anz5bis9(  4) ist gleich 0
14        // anz5bis9(  5) ist gleich 1
15        // anz5bis9(  9) ist gleich 1
16
17        ...
18     }
```

Aufgabe 3 (20 Punkte): Betrachten Sie die folgenden Befehle:

```
Object[] ora = new Object[2];
Object[] orb = new Object[2];

ora[0] = ora;
ora[1] = orb;

orb[0] = orb;
orb[1] = ora;
```

Wie sehen die Variablen `ora` und `orb` nach Ausführung dieser Befehle aus?
Stellen Sie die beiden Variablen als Bojen dar, wahlweise in *ausführlicher* (elaborate) oder in *vereinfachter* (abbreviated) Darstellung. Empfohlen wird die *vereinfachte* Darstellung. Zeichnen Sie "richtige graphische Bojen" (ähnlich denen im Papier **JavaBuoys.pdf**), und keine ASCII-Bojen.

Aufgabe 4 (20 Punkte):

4.1. Ungefähr welchen Wert hat der Ausdruck `3*Integer.MAX_VALUE`?

4.2. Für ungefähr *wie viele Zeichen* (genauer: *code points*) hat der Unicode Platz?

4.3. Schreiben Sie eine Klassen-Methode namens `anz` mit beliebig vielen Parametern vom Typ `Object`, die als Ergebnis die *Anzahl der Argumente* liefert, mit denen sie aufgerufen wurde.
Beispiele:

```
anz()               ist gleich 0
anz("ABC")          ist gleich 1
anz("A", 17, true) ist gleich 3
```

4.4. Genau welche *Tiefe* muss ein binärer Baum *mindestens* haben, wenn er 1 Milliarde Knoten enthält?

4.5. Ungefähr *wie viele Knoten* kann ein binärer Baum der Tiefe 60 *höchstens* haben?

4.6. Genau wie viele Kanten kann ein einfacher Graph mit 20 Ecken höchstens haben? Geben Sie die betreffende Formel an, aber zusätzlich auch "die fertig ausgerechnete Zahl" (z.B. 17 oder 1250 oder so ähnlich).

4.7. Genau wie viele Ecken muss ein einfacher, zusammenhängender Graph mit 200 Kanten *mindestens* haben?

4.7.b Genau wie viele Ecken kann ein einfacher, zusammenhängender Graph mit 200 Kanten höchstens haben?

Aufgabe 5 (20 Punkte): Ganz ähnlich wie in der Datei `SchrittFunktionenA.java` soll auch für diese Aufgabe gelten:

1. Die Methoden `alg53` und `alg54` repräsentieren Algorithmen.
2. Was die Algorithmen genau machen, spielt hier keine Rolle.
3. Der `int`-Parameter `N` gibt die Größe des Problems an, welches bearbeitet wird.
4. Jeder Aufruf der Methode `schritt()` kann als **ein Schritt** gewertet werden. Der Zeitbedarf aller anderen Befehle (z.B. `i++` und `j++` etc.) kann vernachlässigt werden.

5.1. Betrachten Sie den folgenden Algorithmus `alg53`:

```
static void alg53(final int N) {
   for (int i=1; i<=N; i++) {
      for (int j=1; j<=6*i; j++) {
         schritt();
      }
   }
} // alg53
```

Geben Sie eine Schritt-Funktion `stp53` für den Algorithmus `alg53` an.
Wie im SU erläutert wurde, muss diese Schritt-Funktion folgende Spezifikation realisieren:

```
static int stp53(final int N) {
   // Wie viele Schritte muss der Algorithmus alg53 ausfuehren, um
   // ein Problem der Groesse N zu bearbeiten? Diese Methode liefert
   // die Antwort.
   ...
}
```

5.2. Betrachten Sie den folgenden Algorithmus `alg54`:

```
static void alg54(final int N) {
   for (int i=1; i<=2*N; i++) {
      for (int j=1; j<=3*N; j++) {
         for (int k=1; k<=5; k++) {
            schritt();
         }
      }
   }

   for (int i=1; i<=N; i++) {
      for (int j=1; j<=5*N; j++) {
         schritt();
      }
   }
} // alg54
```

Geben Sie eine Schritt-Funktion `stp54` für den Algorithmus `alg54` an.
Wie im SU erläutert wurde, muss diese Schritt-Funktion folgende Spezifikation realisieren:

```
static int stp54(final int N) {
   // Wie viele Schritte muss der Algorithmus alg54 ausfuehren, um
   // ein Problem der Groesse N zu bearbeiten? Diese Methode liefert
   // die Antwort.
   ...
}
```

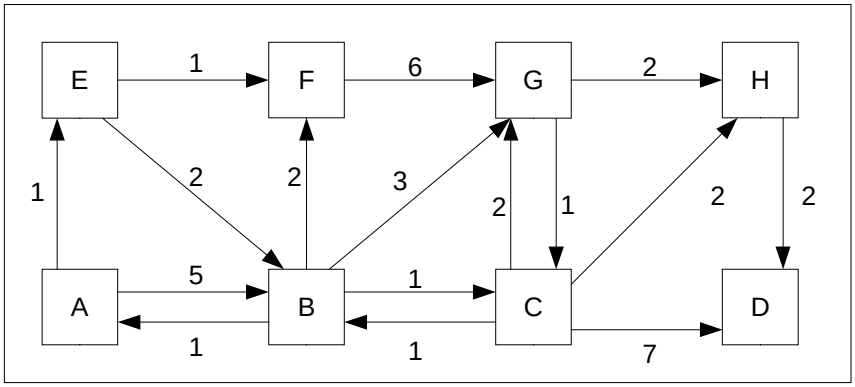
Beurteilung der Klausur:

Punkte↓	
Aufgabe 1:	Note:
Aufgabe 2:	Datum: 30.03.2016
Aufgabe 3:	
Aufgabe 4:	
Aufgabe 5:	
Summe:	

Korrigierte Beurteilung der Klausur:

Punkte↓	
Aufgabe 1:	Note:
Aufgabe 2:	Datum:
Aufgabe 3:	
Aufgabe 4:	
Aufgabe 5:	
Summe:	

Lösung 1 (20 Punkte): Betrachten Sie den folgenden Graphen:



1.1. Berechnen Sie nach dem Dijkstra-Algorithmus die kürzesten Wege vom Knoten A zu allen anderen Knoten, indem Sie die folgende Tabelle ausfüllen (wie im SU und den Ü behandelt):

besucht	**Knoten**		**A 0**	**E 1**	**F 2**	**B 3**	**C 4**	**G 6**	**H 6**	**D 8**	**Min**
1.	A	0									0
4.	B	inf	5	3							3
5.	C	inf				4					4
8.	D	inf					11		8		8
2.	E	inf	1								1
3.	F	inf		2							2
6.	G	inf			8	6	6#				6
7.	H	inf					6	8#			6

1.2. Über welche Knoten führt der kürzeste Weg von A nach D?

A - E - B - C - H - D, Länge 1+2+1+2+2 gleich 8

Lösung 2 (20 Punkte):

```
1      static int anz5bis9(final int N) {
2         // Verlaesst sich darauf, dass N groesser oder gleich 0 ist.
3         // Liefert die Antwort auf folgende Frage: Wenn man N als
4         // 10-er-Zahl (Dezimalzahl) darstellt, wie viele Ziffern sind
5         // dann groesser als 4?
6
7         int anz = (N%10>4) ? 1 : 0;
8         if (N < 10) return anz;
9         return anz + anz5bis9(N/10);
10     }
```

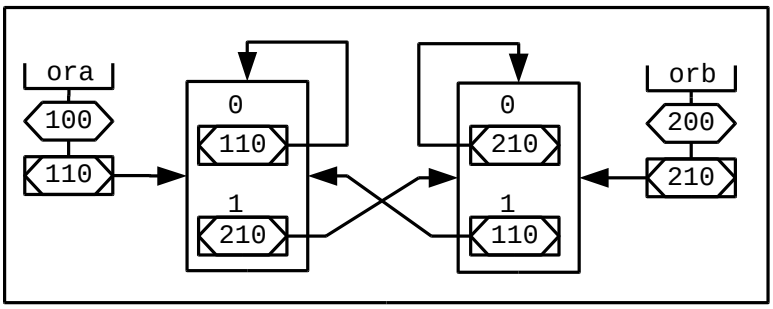
Lösung 3 (20 Punkte): Betrachten Sie die folgenden Befehle:

```
Object[] ora = new Object[2];
Object[] orb = new Object[2];

ora[0] = ora;
ora[1] = orb;

orb[0] = orb;
orb[1] = ora;
```

Wie sehen die Variablen `ora` und `orb` nach Ausführung dieser Befehle aus?
Stellen Sie die beiden Variablen als Bojen dar, wahlweise in *ausführlicher* (elaborate) oder in *vereinfachter* (abbreviated) Darstellung. Empfohlen wird die *vereinfachte* Darstellung. Zeichnen Sie "richtige graphische Bojen" (ähnlich denen im Papier **JavaBuoys.pdf**), und keine ASCII-Bojen.



Lösung 4 (20 Punkte)

4.1. Ungefähr welchen Wert hat der Ausdruck `3*Integer.MAX_VALUE`?

2.15 Milliarden (oder: Integer.MAX_VALUE)

4.2. Für ungefähr *wie viele Zeichen* (genauer: *code points*) hat der Unicode Platz?

1 Million (genau: 1.114.112)

4.3. Schreiben Sie eine Klassen-Methode namens `anz` mit beliebig vielen Parametern vom Typ `Object`, die als Ergebnis die Anzahl der Argumente liefert, mit denen sie aufgerufen wurde.
Beispiele:

```
anz()              ist gleich 0
anz("ABC")         ist gleich 1
anz("A", 17, true) ist gleich 3
```

```
static int anz(Object... or) {
   return or.length;
}
```

4.4. Genau welche *Tiefe* muss ein binärer Baum mindestens haben, wenn er 1 Milliarde Knoten enthält?

Die Tiefe 30

4.5. Ungefähr *wie viele Knoten* kann ein binärer Baum der Tiefe 60 höchstens haben?

Ungefähr 1 Trillion Knoten

4.6. Genau wie viele Kanten kann ein einfacher Graph mit 20 Ecken höchstens haben? Geben Sie die betreffende Formel an, aber zusätzlich auch "die fertig ausgerechnete Zahl" (z.B. 17 oder 1250 oder so ähnlich).

Formel: N*(N-1)/2, Ergebnis: 190

4.7. Genau wie viele Ecken muss ein einfacher, zusammenhängender Graph mit 200 Kanten mindestens haben?

Mindestens 21 Ecken (denn ein einfacher, zusammenhängender Graph mit 20 Ecken kann höchstens 20*19/2 gleich 190 Kanten haben, aber ein solcher Graph mit 21 Ecken kann 21*20/2 gleich 210 Kanten haben).

4.7.b Genau wie viele Ecken kann ein einfacher, zusammenhängender Graph mit 200 Kanten höchstens haben?

Höchstens 201 Ecken

Lösung 5 (20 Punkte):

5.1.

```
static int stp53(final int N) {
   // Wie viele Schritte muss der Algorithmus alg53 ausfuehren, um
   // ein Problem der Groesse N zu bearbeiten? Diese Methode liefert
   // die Antwort.

   return 3*N*N + 3*N;
}
```

Herleitung (nach Formel: ErsteZahl + LetzteZahl * AnzahlPaare):

i	1	2	3	...	N
Anz. Schritte	6	12	18	...	6*N

```
(6 + 6*N) * N/2 = 3*N + 3*N*N
```

5.2.

```
static int stp54(final int N) {
   // Wie viele Schritte muss der Algorithmus alg54 ausfuehren, um
   // ein Problem der Groesse N zu bearbeiten? Diese Methode liefert
   // die Antwort.

   return 35*N*N;
}
```