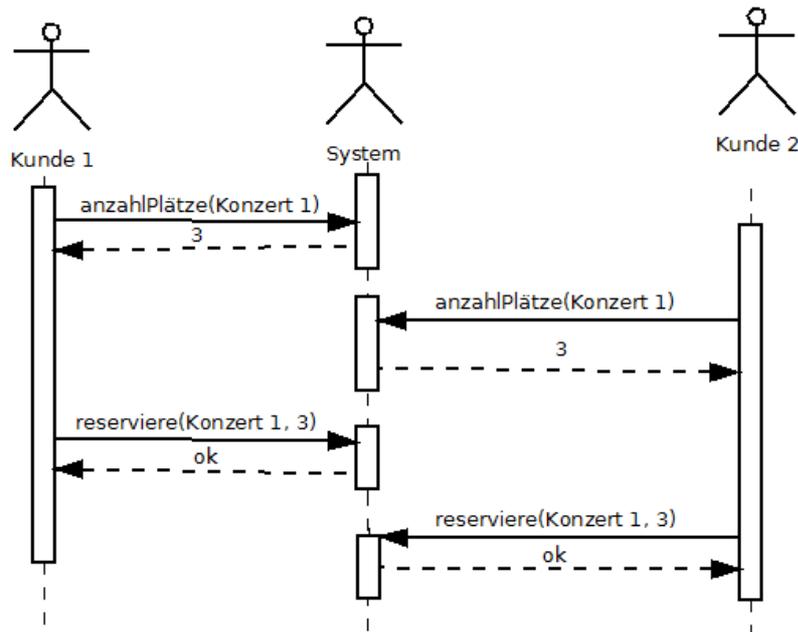


1a

5



1b Das Monitorkonzept:

3

Mit **Monitoren** können Verklemmungen verhindert werden.

Mit Monitoren lassen sich geschützte Programmabschnitte definieren, die immer nur von einem Prozess gleichzeitig ausgeführt werden dürfen.

In diesem Beispiel müssen die beiden Methoden `anzahlPlätze` und `reserviere` durch eine einzige Methode ersetzt werden, die beides leistet. Die Methode wird durch einen Monitor geschützt.

In Java geschieht dies durch das Schlüsselwort `synchronized`.

2 Unter einer **Verklemmung** versteht man in der Informatik einen Zustand, bei dem eine zyklische Wartesituation zwischen mehreren Prozessen auftritt, wobei jeder beteiligte Prozess auf die Freigabe von Betriebsmitteln wartet, die ein anderer beteiligter Prozess bereits exklusiv belegt hat.

Beispiel:

Die Ampeln A1 und A4 sind grün, A2 und A3 sind rot. Wenn aus beiden Richtungen sehr viele Autos kommen, staut es sich an A2 bzw. A3 so, dass die beiden einspurigen Abschnitte mit Autos belegt sind. A1 und A4 können zwar auf rot schalten, aber A2 und A3 nicht auf grün. Somit hat man den Zustand der Verklemmung.

3a

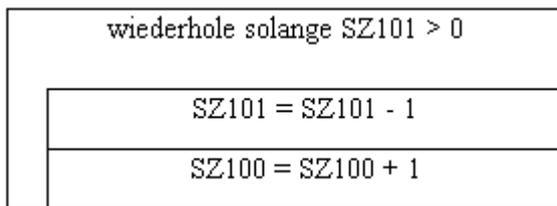
5

	A	PC	N	Z	SZ 100	SZ 101
Startbelegung:	0	0			17	2
0: load 101	2	2	0	0	17	2
2: jmp 6	2	6	0	0	17	2
6: dec	1	8	0	0	17	2
8: store 101	1	10	0	0	17	1
10: load 100	17	12	0	0	17	1
12: inc	18	14	0	0	17	1
14: store 100	18	16	0	0	18	1
16: jmp 0	18	0	0	0	18	1
0: load 101	1	2	0	0	18	1
2: jmp 6	1	6	0	0	18	1
6: dec	0	8	0	0	18	1

8: store 101	0	10	0	1	18	0
10: load 100	18	12	0	1	18	0
12: inc	19	14	0	1	18	0
14: store 100	19	16	0	1	19	0
16: jmp 0	19	0	0	1	19	0
0: load 101	0	2	0	1	19	0
2: jmp 6	0	6	0	1	19	0
4: hold	0	4	0	1	19	0

(Hinweis: Die ausführliche Angabe einer Zustandstabelle wie oben ist nicht explizit gefordert.)
Dementsprechend ist der Endzustand:

A	PC	N	Z	100	101
0	4	0	1	19	0



3b

```

0:   load 101
2:   jmp 18
4:   jmpz 30
6:   inc
8:   store 101
10:  load 100
12:  dec
14:  store 100
16:  jmp 0
18:  dec
20:  store 101
22:  load 100
24:  inc
26:  store 100
28:  jmp 0
30:  hold

```

5

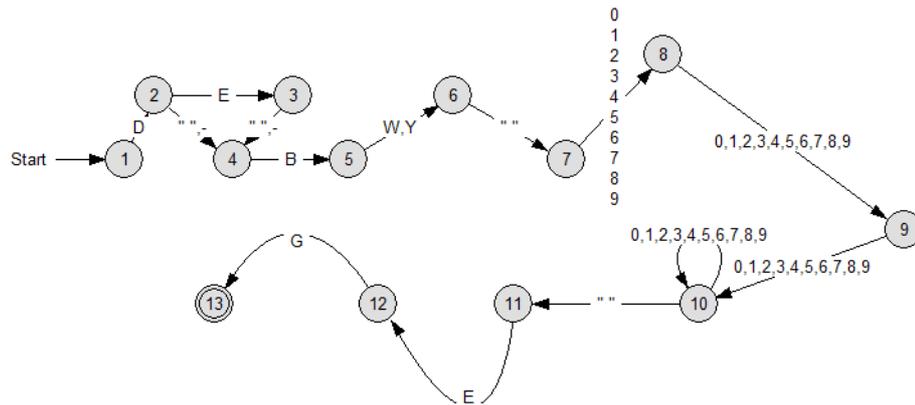
4a Genusstauglichkeitskennzeichnung = a [a] [a] b [c] n n n {n} “ “ a a
a = “A” | “B” | ... | “Z”
b = - | “ “
c = a [a] “ “
n = “0” | “1” | ... | “9”

6

AAA – AB 012345 AA wäre eine syntaktisch korrekte Genusstauglichkeitskennzeichnung.
Diese ist jedoch nicht semantisch korrekt, da AAA offensichtlich keine Landeskennung eines Landes der EU ist.

4b

5



4c class NEA{

7

```

    private int state;

    private void zeichenBearbeiten(char i) {
        switch(state) {
            ...
            case 10:
                switch(i) {
                    case '0':
                    case '1':
                        ...
                    case '8':
                    case '9': state = 10;
                        break;
                    case ' ': state = 11;
                        break;
                    default: state = -1;
                }
            break;
            ...
        }
    }
}

```

```

boolean testen(String gtk) {
    state = 1;
    int z = 0;
    while(z < gtk.laenge()) {
        zeichenBearbeiten(gtk.zeichenAn(z));
        z = z + 1;
    }
    return(state == 13);
}
}

```

40