Informatik 2017 – I

1.)

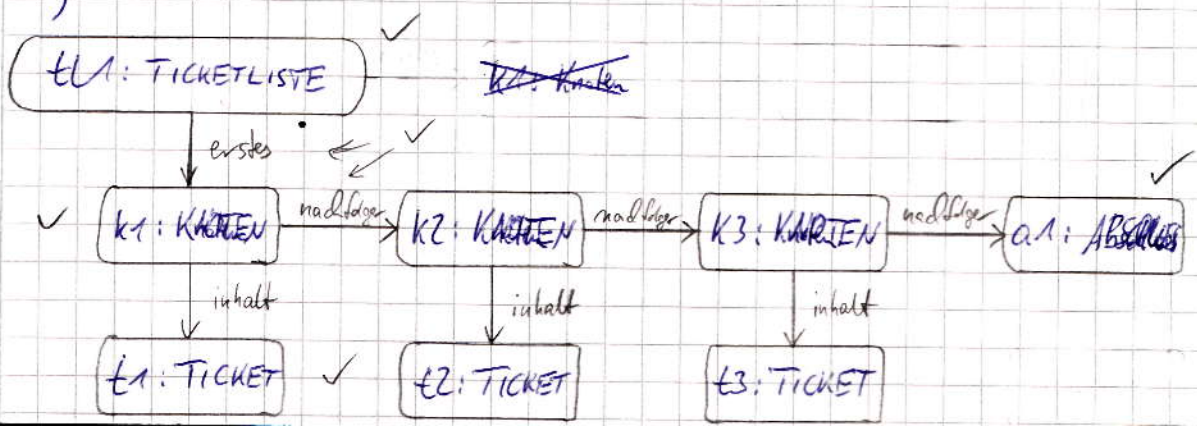
```
public class TICKET {
    private String kennzeichen;
    private String telefonnr;
    private int parkende;
    public TICKET (String k, String t, int e) {
        kennzeichen = k;
        telefonnr = t;
        parkende = e;
    }
    public int restzeitgeben (int uhrzeit) {
        return parkende - uhrzeit;
    }
}
```

5

2.) Im Parksystem wird der Parkwunsch registriert, indem Kennzeichen, Telefonnr und gewünschte Parkdauer übermittelt werden.
Das Parksystem holt sich von der Uhr die aktuelle Uhrzeit und erstellt mit dem Kennzeichen, Telefonnr. und der errechneten Endzeit ein neues Ticket, das anschließend in die Ticketliste eingefügt wird.
Damit ist die Registrierung abgeschlossen.

4

3a)



5

b) TICKETLISTE:

```
public void einfuegen (TICKET ticket) {
    erstes = new KNOTEN (ticket, erstes);
}
public TICKET suchen (String kennzeichen) {
    return erstes.suchen (kennzeichen);
}
```

LISTENELEMENT:

```
abstract TICKET suchen (String kennzeichen);
```

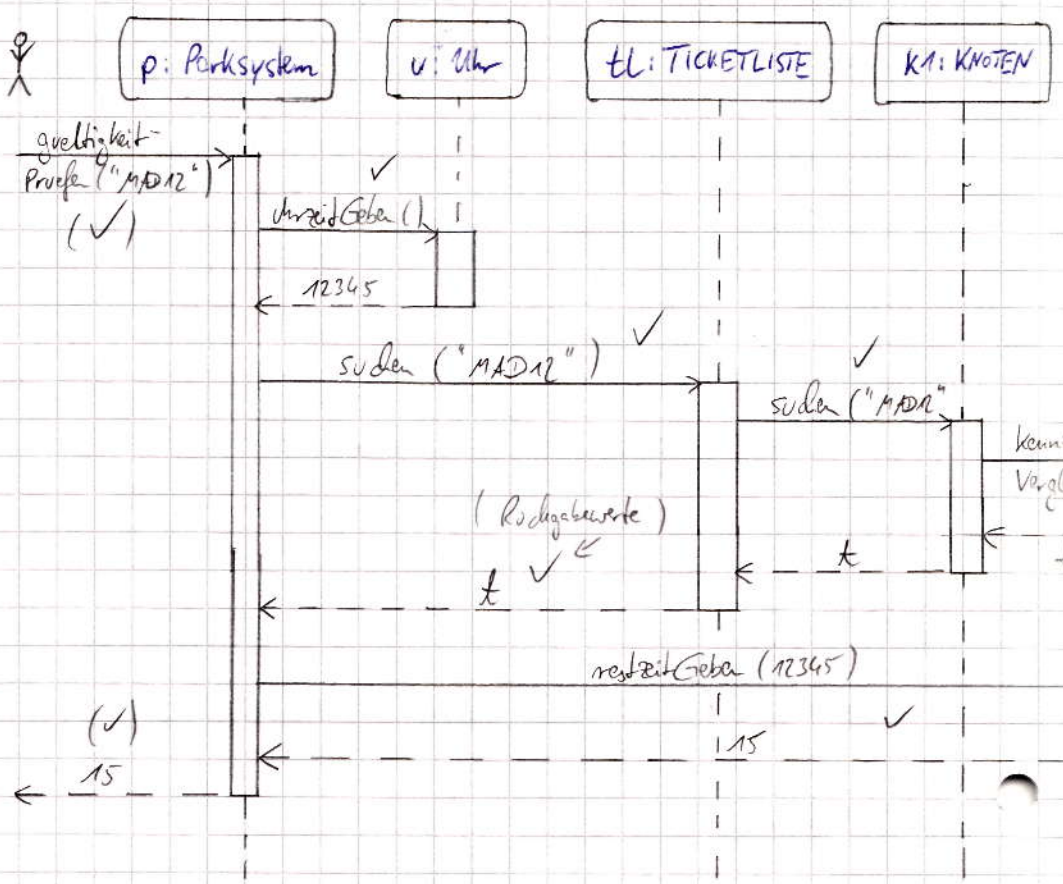
KNOTEN:

```
public KNOTEN (TICKET t; LISTENELEMENT n) {
    inhalt = t;
    nachfolger = n;
}
public TICKET suchen (String kennzeichen) {
    if (inhalt.kennzeichenVergleichen (kennzeichen))
        return this.inhalt;
    else
        return nachfolger.suchen (kennzeichen);
}
```

ABSCHLUSS:

```
public TICKET suchen (String kennzeichen) {
    return null;
}
```

c)



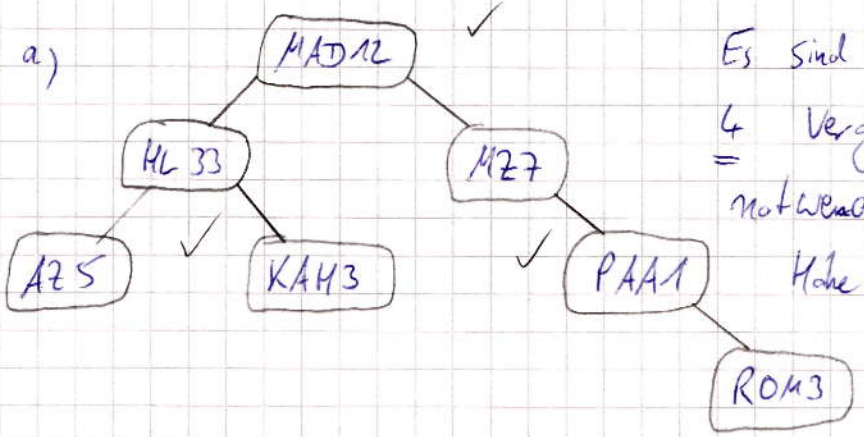
8

d)

```
gueltigkeitPruefen (Kennzeichen)
    uz = u.uhrzeitGeben();
    t = tl.suchen (Kennzeichen)
    wenn t = null ✓ gib -99999 ✓ zurück
    sonst gib t.restzeitGeben (uz) ✓ zurück
```

3

4.) a)



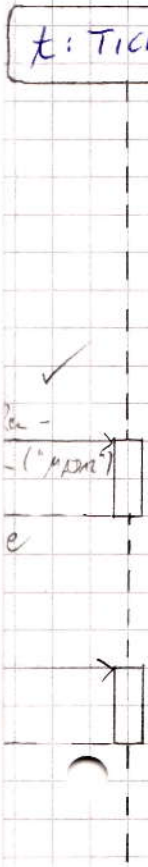
Es sind maximal 4 Vergleiche ✓ = notwendig, da Höhe 4.

b) Obergrenze: Liste ✓, da im ungünstigsten Fall bis zum letzten Element verglichen werden muss

⇒ 250 Vergleiche ✓

Optimal: (balancierter) Binärbaum ✓, da dieser nur $\lceil \log_2(250) \rceil = 8$ ✓ Ebenen ✓ hat sind auch maximal 8 ✓ Vergleiche notwendig.

6



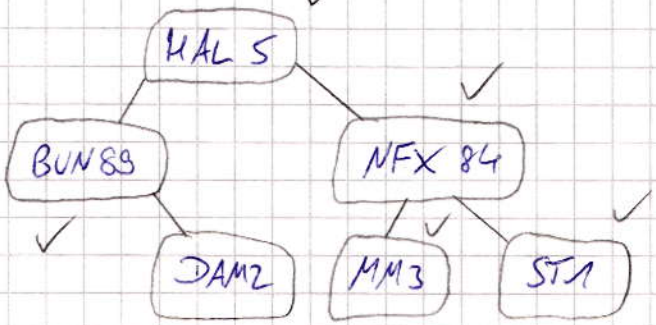
c) Jeder Knoten hat nur einen linken und einen rechten Nachfolger. ✓ ✓ 2

d) KNOTEN:

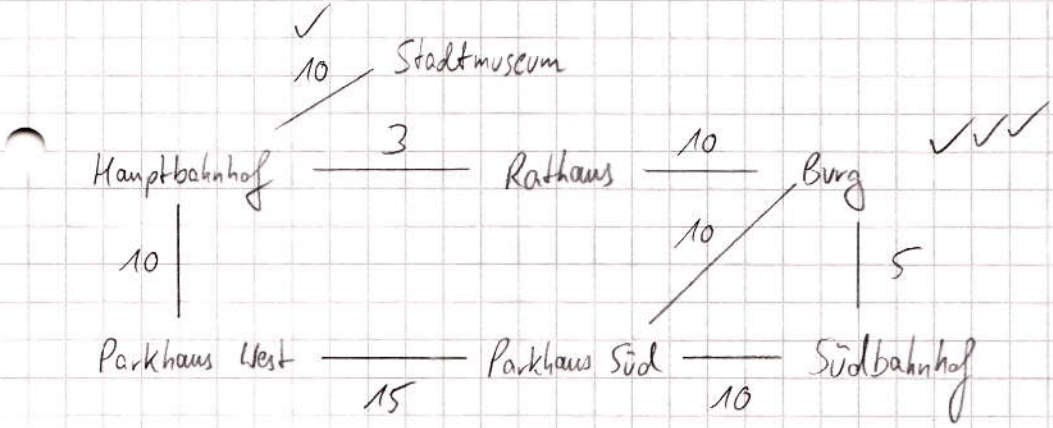
```
abgelaufene zeigen () {
    linker Nachfolger. abgelaufene zeigen ()
    wenn ticket abgelaufen dann
        ticket ausgeben
    rechter Nachfolger. abgelaufene zeigen ()
}
```

4

e) 5



5.) Ungerichteter, gewichteter (, zusammenhängender) Graph 9

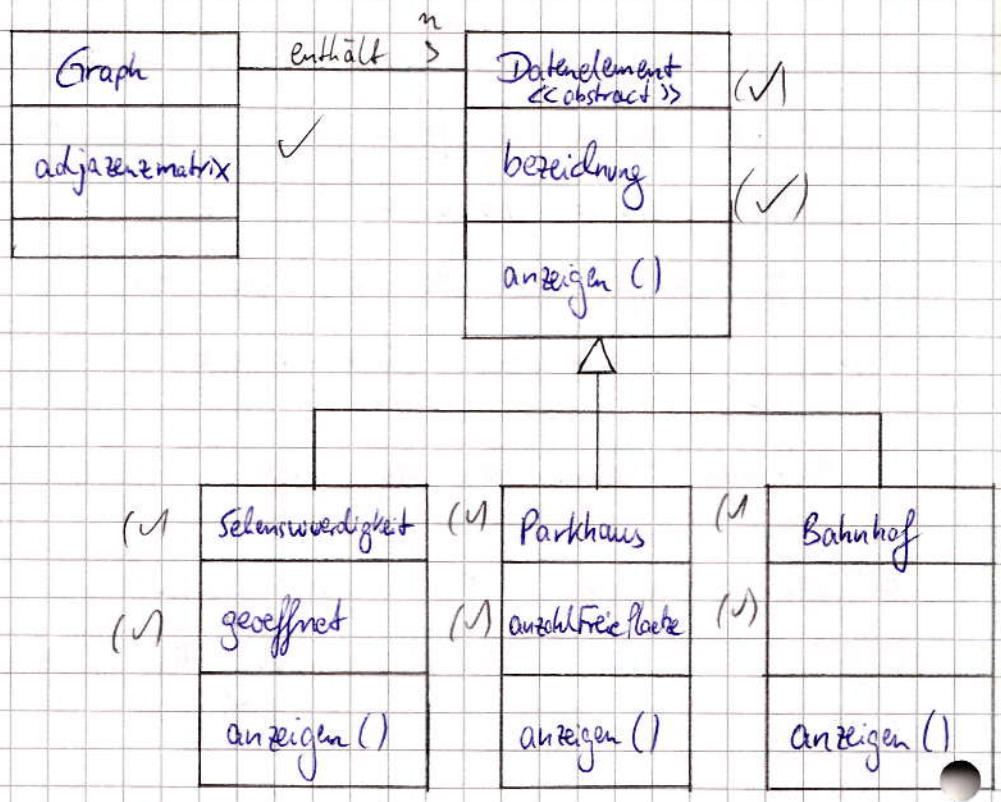


	Museum	Hbf	PH W	PH S	Rathaus	Burg	Süd-Bhf
Museum	0	10	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
Hbf	10	0	10	$-\infty$	3	$-\infty$	$-\infty$
PH W	$-\infty$	10	0	15	$-\infty$	$-\infty$	$-\infty$
PH S	$-\infty$	$-\infty$	15	0	$-\infty$	10	10
Rathaus	$-\infty$	3	$-\infty$	$-\infty$	0	10	$-\infty$
Burg	$-\infty$	$-\infty$	$-\infty$	10	10	0	5
Süd-Bhf	$-\infty$	$-\infty$	$-\infty$	10	$-\infty$	5	0

Bogen 2

5

b)



9

c) wiederhole für alle Knoten k {
wenn k vom Startknoten ks erreichbar ist {
gib Bezeichnung von k und Kantengewicht aus
wiederhole für alle Knoten k2 {
wenn k2 von k erreichbar und ungleich ks ist
gib Bezeichnung von k2 und die Kantengewichte (ks - k) + (k - k2) + 5 aus

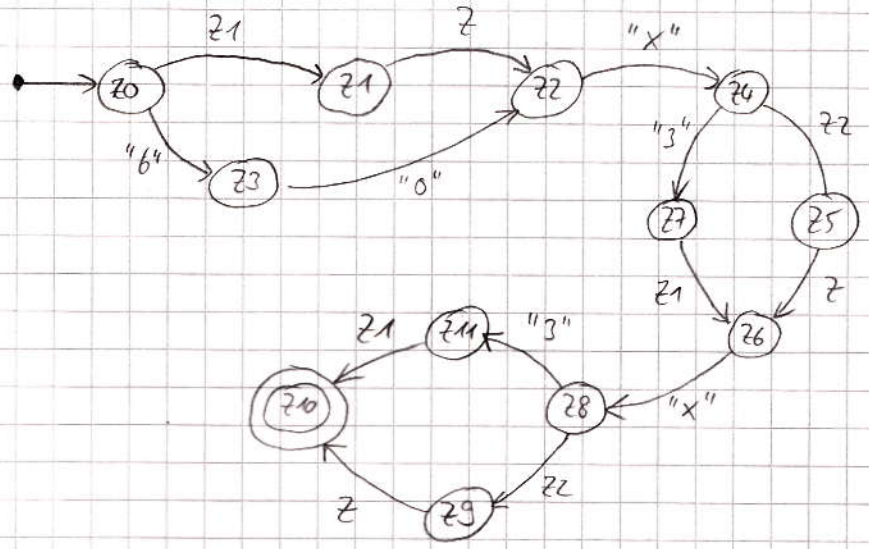
} } } } }

Informatik 2017 – III

1a) $G = (\{$Größe, Länge, Breite, Höhe, Z1, Z2, Z$\}$, ✓
$\{$"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "x"$\}$, ✓
P, Größe) 5

P:
Größe = Länge "x" Breite "x" Höhe ✓
Länge = Z1 Z | "60" ✓
Breite = Z2 Z | "3" Z1
Höhe = Breite
Z1 = "0" | "1" | "2" | "3" | "4" | "5" ✓
Z2 = "1" | "2" | "0"
Z = Z1 | "6" | "7" | "8" | "9"

b) ✓✓✓✓ 4



c) Z0 → Z1 → Z2 → Z4 → Z5 → Z6 → Z8 → Z9 → Z10 ✓✓ 2
4 7 x 0 8 x 1 5

d) Syntax: Ein Wort lässt sich durch die Produktionsregeln ✓
einer Grammatik erzeugen. Bsp.: siehe c)
Semantik: Bedeutung des Wortes. Im Bsp.: Länge, ✓ 2
Breite und Höhe des Pakets.

2.) a)

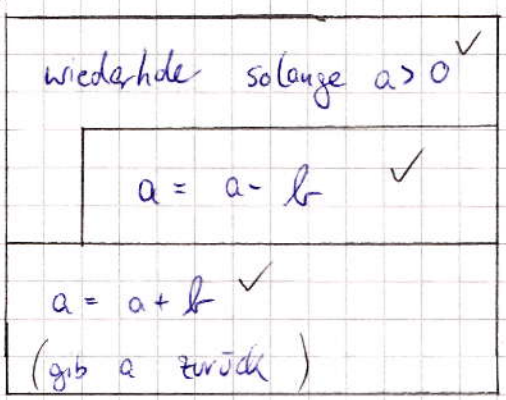
201	202	AK		
6	17	17		LOAD 202
6	17	11	✓	SUB 201
6	17	11		JLT 15
6	11	11		STORE 202
6	11	11		JMP 11
6	11	5	✓	SUB 201
6	181	5		JLT 15
6	5	5		STORE 202
6	5	5		JMP 11
6	5	-1		SUB 201
6	5	-1	✓	JLT 15
6	5	5		LOAD 202

3

b) Der Rückgabewert ist der Rest bei Division der Zahl ✓
in 202 durch die Zahl in 201.
Mögliche Prüfziffern: 0; ...; 5 ✓

2

c) ✓



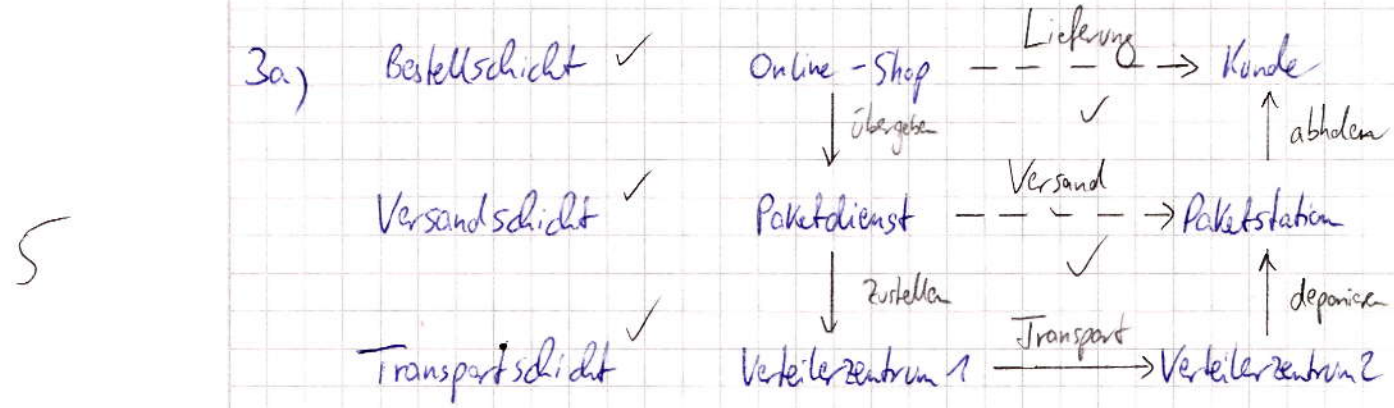
4

d)

1	LOAD	201	✓
2	JGT	10	✓
3	LOAD	-1	✓
4	JMP	16	✓

3

3a)



5

b) Vorteil: Komplexe Arbeitsabläufe ✓ werden besser
strukturiert ✓

2

4.) Während die Reservierung von Kunde 1 ✓ angenommen wurde, jedoch die Anzahl der freien Fäder noch nicht ✓ reduziert wurde, reserviert Kunde 2. ✓
Abhilfe schafft ein Monitor ✓, der nur eine exklusive Nutzung der kritischen Methode erlaubt. ✓

4

5.) Maximale Dauer bei brute force: $0{,}2 \cdot 10^{-3}\,s \cdot 10^5 = \underline{\underline{20\,s}}$ ✓

4

Mögliche Maßnahmen: 1.) höhere Passwortlänge ✓
2.) Vergrößerung des verwendeten Alphabets ✓