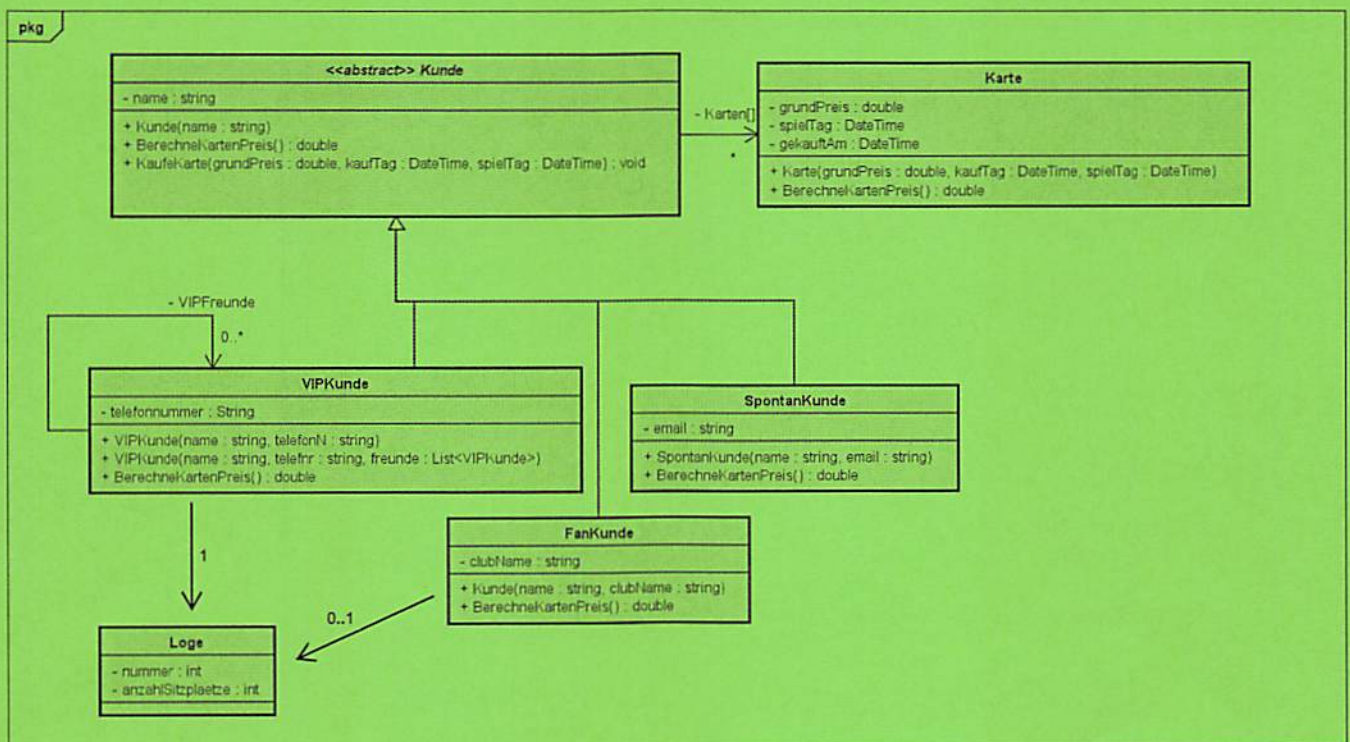


**Lösungsvorschläge:** Lösungsvorschläge sind in der Regel Vorschläge der einreichenden Schulen; sie sind im Wortlaut nicht bindend. Anderslautende, aber zutreffende Antworten sind ebenfalls als richtig zu werten.

**Nur für die Hand  
des Prüfers!  
Faktor**

**Aufgabe 1 SAE****2****1.1**

1.2

```
class Karte
{
    private DateTime _spielTag;
    private DateTime _gekauftAm;
    private double _grundPreis;

    public double BerechneKartenpreis ()
    {
        int diffTage = (_spielTag - _gekauftAm).Days;
        if (diffTage > 30)
            return _grundPreis * 0.90;
        else
            return _grundPreis;
    }

    public Karte(double grundPreis,DateTime kaufTag, DateTime spieltag)
    {
        _grundPreis = grundPreis;
        _gekauftAm=kaufTag;
        _spieltag = spieltag;
    }
}

abstract class Kunde
{
    private string _name;
    protected List<Karte> _karten;

    public Kunde(string name)
    {
        _name = name;
        _karten = new List<Karte>();
    }

    public virtual double BerechneKartenpreis()
    {
        double preis = 0.0;

        foreach (Karte karte in _karten) //oder als Linq
        {
            preis += karte.BerechneKartenpreis();
        }

        return preis;
    }

    public void KaufeKarte(double grundPreis,DateTime kaufTag, DateTime spielTag)
    {
        Karte k = new Karte(grundPreis, kaufTag, spielTag);
        _karten.Add(k);
    }
}
```



```
}

class VIPKunde : Kunde
{
    string _telefonNummer;
    private List<VIPKunde> _vipFreunde;

    public VIPKunde(string name, string telefonnr)
        : base(name)
    {
        _vipFreunde = null;
        _telefonNummer = telefonnr;
    }

    public VIPKunde(string name, string telefonnr, List<VIPKunde> freunde)
        : base(name)
    {
        _vipFreunde = freunde;
        _telefonNummer = telefonnr;
    }

    public override double BerechneKartenpreis()
    {
        double kartenPreis;
        kartenPreis = base.BerechneKartenpreis();

        if (_vipFreunde.Count > 1)
        {
            kartenPreis *= 0.95;
        }

        return kartenPreis;
    }
}
```

## 1.3

```
//Testprogramm
class Program
{
    static void Main(string[] args)
    {
        VIPKunde freund1 = new VIPKunde("Toni Freund", "02342 3333333");
        VIPKunde freundin2 = new VIPKunde("Tina Freundinger", "0432 3111111");

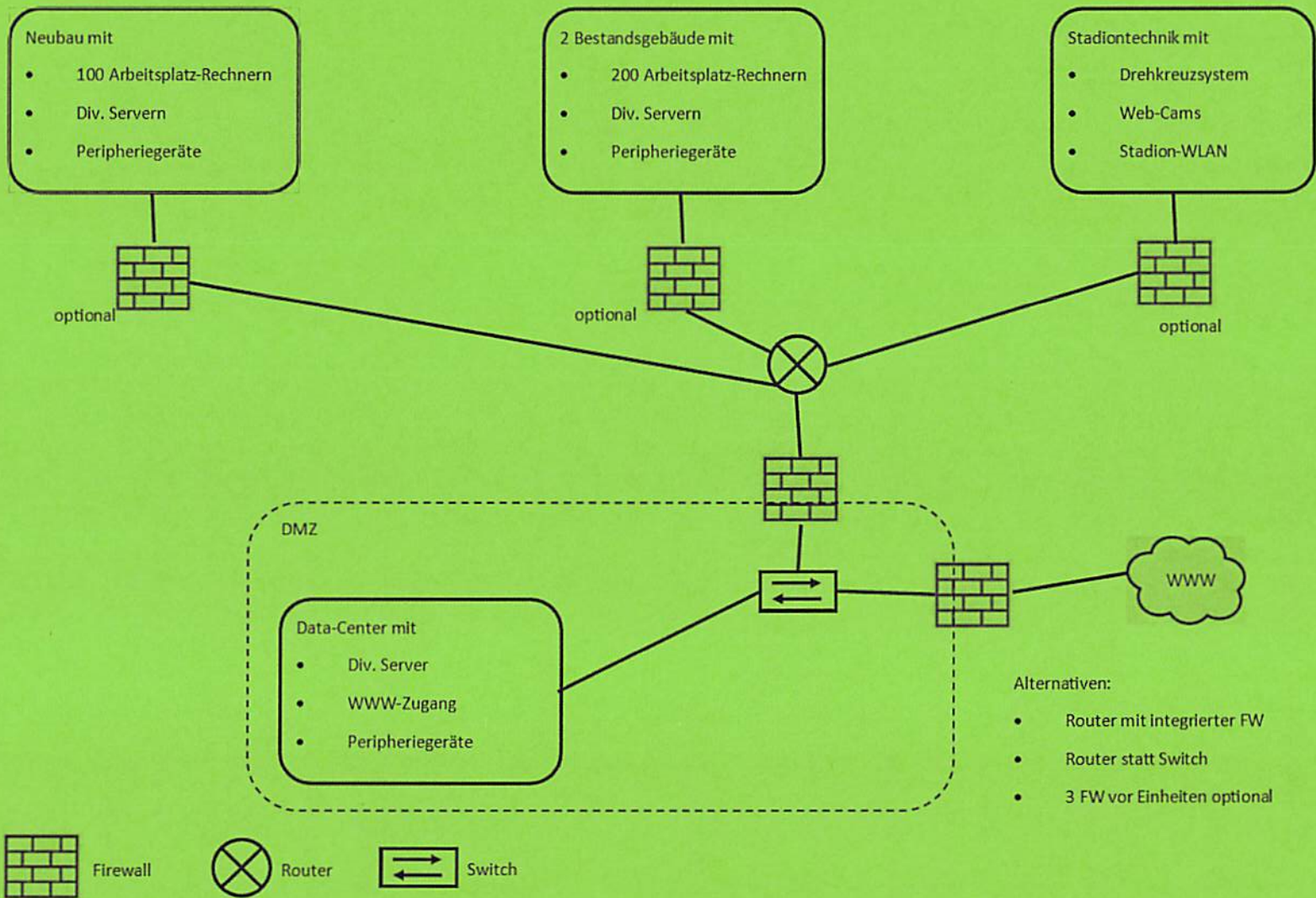
        VIPKunde klopp = new VIPKunde("Johannes Klopp", "01234 4391282",
            new List<VIPKunde> { freund1, freundin2 });

        klopp.KaufeKarte(24.5, new DateTime(2020, 6, 1), new DateTime(2020, 7, 4));
        klopp.KaufeKarte(24.5, new DateTime(2020, 6, 1), new DateTime(2020, 6, 6));

        double kartenPreis = klopp.BerechneKartenpreis();
        Console.WriteLine("Die Karten kosten: " + kartenPreis);
        Console.ReadLine();
    }
}
```

## Aufgabe 2 ITS

2.1





## 2.2.1

Netz: 10.3.3.0	0000 0000	
SNM: 255.255.255.248	1111 1000	Fehler 2: Subnetzmaske zu klein. Es liegen nur DK9, DK10 und der Server im selben SN.
DK1: 10.3.3.1	0000 0001	
DK2: 10.3.3.2	0000 0010	
DK3: 10.3.3.3	0000 0011	
DK4: 10.3.3.4	0000 0100	
DK5: 10.3.3.5	0000 0101	
DK6: 10.3.3.6	0000 0110	
DK7: 10.3.3.7	0000 0111	
DK8: 10.3.3.8	0000 1000	Fehler1: Ist keine Host-Adresse sondern Subnetzadresse. Host darf keine SNA erhalten.
DK9: 10.3.3.9	0000 1001	
DK 10: 10.3.3.10	0000 1010	
Server: 10.3.3.11	0000 1011	

DK7: IP-Adresse ist Broadcast-Adresse  
DK8: IP-Adresse ist Netzwerkadresse

## 2.2.2 Subnetzmaske „vergrößern“ auf 255.255.255.240

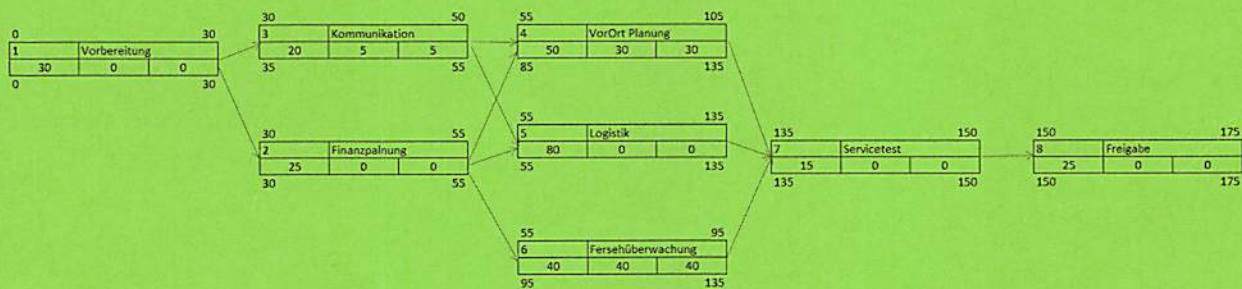
## Aufgabe 3 BWL

1

## 3.1

Vorgangs-Nr.	Beschreibung	Vorgänger	Aufwand in Personentagen	Anzahl zur Verfügung stehende Personen	Dauer in Arbeitstagen
1	Vorbereitung	---	60	2	30
2	Finanzplanung	1	50	2	25
3	Kommunikationsplanung	1	40	2	20
4	Vor-Ort-Planung	2, 3	100	2	50
5	Logistik	2, 3	80	1	80
6	Fernsehübertragung	2	40	1	40
7	Servicetest	4, 5, 6	30	2	15
8	Freigabe	7	50	2	25

## 3.2



## 3.3

Das Projekt kann nicht termingerecht fertig gestellt werden. Das Paket liegt zwar nicht auf dem kritischen Pfad, doch die Verzögerung ist größer als der freie Puffer.