

Liebe Schüler,

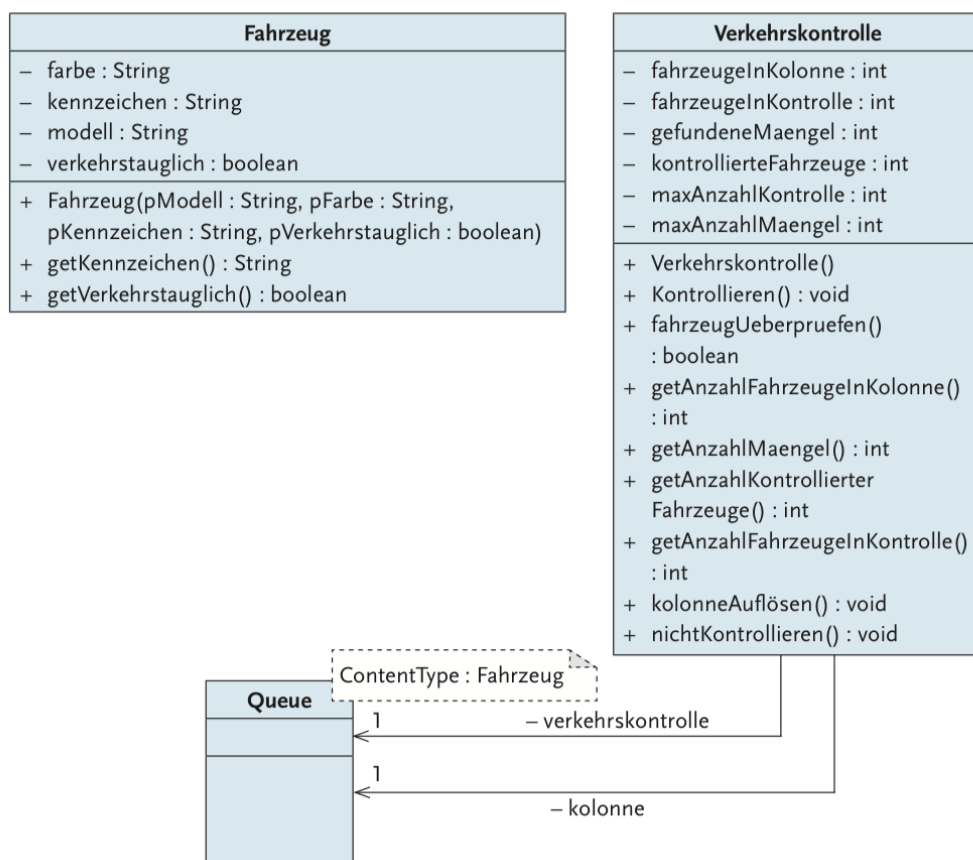
hier die Lösungen zu den Buch-Aufgaben vom 13.10. Bitte vergleicht eure Lösungen mit den Musterlösungen und verbessert eure Lösungen.

Die anderen Aufgaben zum niedersächsischen Abitur werden wir am Montag besprechen.

### 3. Verkehrskontrolle

a) Bei einer Verkehrskontrolle gilt das FIFO-Prinzip (First In First Out). Es wird zuerst das Auto kontrolliert, das als erstes herausgewunken wurde. Alle folgenden zu kontrollierenden Fahrzeuge reihen sich dahinter ein. Ist die Kontrolle des ersten Fahrzeugs beendet, verlässt es die Kontrolle und das nächste kann kontrolliert werden. Das Fahrzeug, das als letztes herausgewunken wurde, wird auch als letztes kontrolliert.

#### b) Implementationsdiagramm Verkehrskontrolle



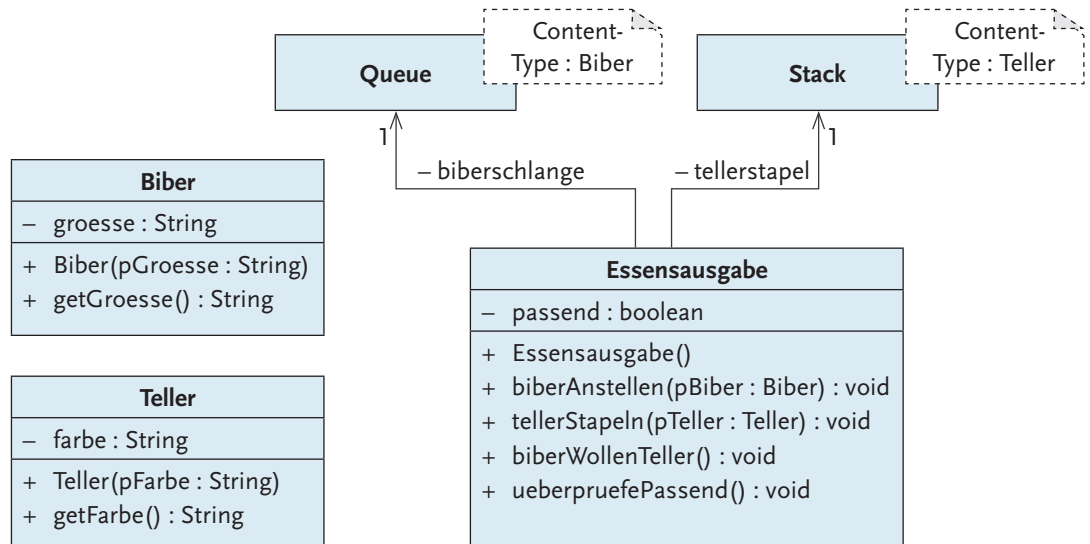
## SB S. 68

## 3. Aus dem Informatik Biber 2010

## a) Biber und Teller

- A) Tellerstapel und Biberschlange passen zueinander.  
 B) Tellerstapel und Biberschlange passen zueinander.  
 C) Tellerstapel und Biberschlange passen nicht zueinander. Der erste Biber benötigt, ebenso wie der zweite, einen flachen braunen Teller. Auf dem Stapel liegen aber zu oberst zwei tiefe grüne Teller.  
 D) Tellerstapel und Biberschlange passen zueinander.

## b)



## c) Klasse Teller

```

public class Teller
{
    private String farbe;
    /**
     * Konstruktor für Objekte der Klasse Teller.
     * @param pFarbe gruen oder braun
     */
    public Teller(String pFarbe)
    {
        farbe = pFarbe;
    }

    /**
     * @return gruen oder braun
     */
    public String getFarbe()
    {
        return farbe;
    }
}
  
```

## Klasse Biber

```

public class Biber
{
    private String groesse;

    /**
     * Konstruktor für Objekte der Klasse Biber.
     * @param pGroesse klein oder gross
     */
  
```

```

        */
    public Biber(String pGroesse)
    {
        groesse = pGroesse;
    }

    /**
     * @return groesse klein oder gross
     */
    public String getGroesse()
    {
        return groesse;
    }
}

```

### Klasse *Essensausgabe*

```

public class Essensausgabe
{
    private boolean passend;
    private Stack<Teller> tellerstapel;
    private Queue<Biber> biberschlange;

    public Essensausgabe()
    {
        passend = true;
        tellerstapel = new Stack<Teller>();
        biberschlange = new Queue<Biber>();
    }

    public void tellerStapeln(Teller pTeller) {
        tellerstapel.push(pTeller);
    }

    public void biberAnstellen(Biber pBiber) {
        biberschlange.enqueue(pBiber);
    }

    public void biberWollenTeller()
    {
        while(!tellerstapel.isEmpty() && !biberschlange.isEmpty() &&
            passend==true) {
            ueberpruefePassend();
        }

        if (passend==true) {
            System.out.println("Alle Biber haben einen passenden Teller
            bekommen.");
        }
        else {
            System.out.println("Der Tellerstapel passte nicht zur Biber-
            schlange.");
        }
    }
}

```

```

public void ueberpruefePassend() {
    boolean passend;
    Teller aktuellerTeller;
    Biber aktuellerBiber;
    aktuellerTeller = tellerstapel.top();
    aktuellerBiber = biberschlange.front();

    if (!tellerstapel.isEmpty() && !biberschlange.isEmpty()) {

        if ((aktuellerBiber.getGroesse().equals("klein") && aktu-
ellerTeller.getFarbe().equals("gruen")) || aktuellerBiber.getGroesse().
equals("gross") && aktuellerTeller.getFarbe().equals("braun")) {
            System.out.println("Der vorne stehende Biber findet einen
            passenden Teller vor.");
            biberschlange.dequeue();
            tellerstapel.pop();
        }
        else {
            passend=false;
            System.out.println("Der vorne stehende Biber findet
            keinen passenden Teller vor.");
        }
    }
}

```