

# Softwaretechnik

## Hausaufgabenblatt 8

Patrick Gustav Blaneck

Letzte Änderung: 17. Januar 2022

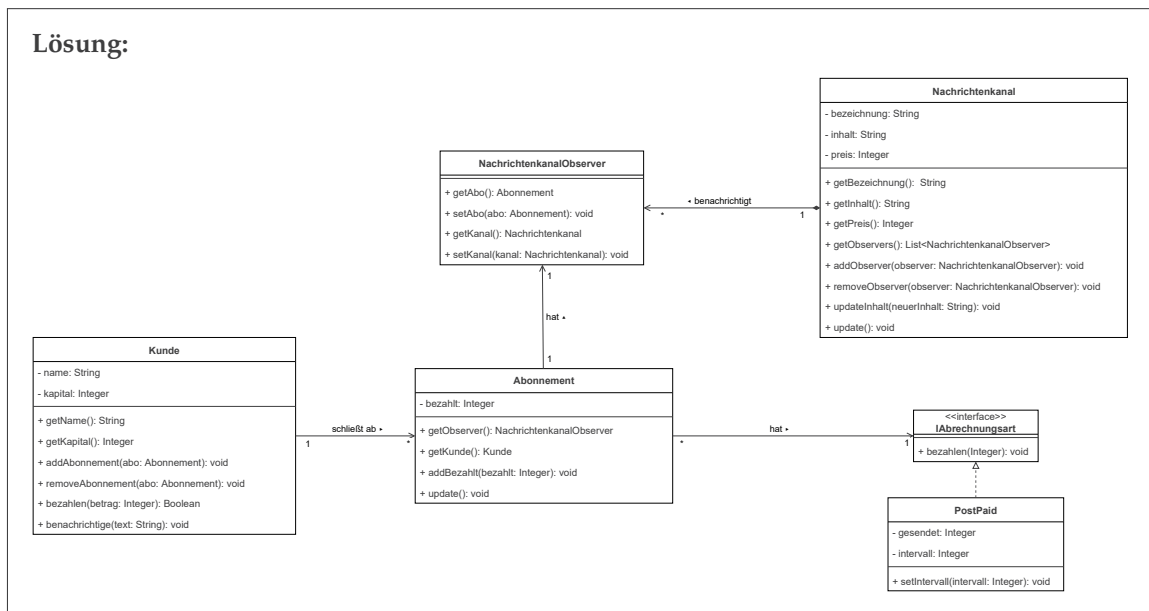
### 1. Nachrichtenverwaltungssystem

Führen Sie die Aufgabe aus dem Praktikum zu Ende.

Zur Erinnerung:

In einem Nachrichtenverwaltungssystem können sich Kunden für unterschiedliche Nachrichtenkanäle interessieren. Dabei kann ein Kunde Nachrichtenkanäle abonnieren und wieder abbestellen. Ändert sich ein Nachrichteninhalt eines Nachrichtenkanals, so werden alle Abonnenten (Kunden) unmittelbar informiert. Der Kunde kann zwischen verschiedenen Abrechnungsarten für seine Abonnements wählen. Für das konkrete Beispiel soll es die Varianten geben, dass für jede Nachricht ein fester Betrag abgerechnet wird und dass man nur jeweils nach jeder dritten Nachricht einen Betrag bezahlen muss. Weitere Abrechnungsarten sollen leicht hinzugefügt werden können.

- (a) Entwerfen Sie ausgehend von den Klassen Nachrichtenkanal und Kunde ein *Klassendiagramm*, das die genannten Forderungen umsetzt. Dabei sollen Sie für die Nachrichten und Kunden das Beobachter-Muster (Observer-Observable-Pattern) und das Strategy-Pattern für die Bezahlmethode einsetzen.



- (b) Geben Sie dann *Implementierungen* aller Klassen an. Beachten Sie, dass für versandte Nachrichten eine Abrechnung erfolgen muss. Binden Sie den Nutzungsdialog (siehe Main.java und Beispiel-Output) mit ein. Mit ihm kann man neue Nachrichten und Kunden anlegen. Außerdem kann der Kunden Nachrichtenkanäle mit einer auswählbaren Abrechnungsart wählen und Informationen über alle Nachrichtenkanäle zusammen mit den angefallenen Abokosten anzeigen. Natürlich kann man auch die Nachrichteninhalte eines Nachrichtenkanals verändern. Der Nutzungsdialog kann nach einigen Schritten wie folgt aussehen, Eingaben sind umrandet (Die Datei Main.java sollte entsprechend ergänzt werden).

*Hinweis: Wie immer ist eine iterative Entwicklung der Lösung sinnvoll, so können Erkenntnisse aus (b) zu Veränderungen der Lösungen von (a) führen!*

<pre> Was wollen Sie? (0) Programm beenden (1) neuen Kunden erstellen (2) neuen Nachrichtenkanal erstellen (3) Kunde abonniert Nachrichtenkanal (4) Nachrichtenkanalstatistik (5) Nachricht eines Nachrichtenkanals verändern 1 Kundenname? Urs Was wollen Sie? (0) Programm beenden (1) neuen Kunden erstellen (2) neuen Nachrichtenkanal erstellen (3) Kunde abonniert Nachrichtenkanal (4) Nachrichtenkanalstatistik (5) Nachricht eines Nachrichtenkanals verändern 3 Welcher Kunde? (1) Ute (2) Uwe (3) Urs 3 Welcher Nachrichtenkanal? (1) Auto (2) Geld 1 Welche Bezahlart (1) pro Nachricht (2) nach jeder 3. Nachricht? </pre>	<pre> 2 Was wollen Sie? (0) Programm beenden (1) neuen Kunden erstellen (2) neuen Nachrichtenkanal erstellen (3) Kunde abonniert Nachrichtenkanal (4) Nachrichtenkanalstatistik (5) Nachricht eines Nachrichtenkanals verändern 5 Welcher Nachrichtenkanal? (1) Auto (2) Geld 1 Neue Nachricht: Fiat Mopster released Ute erhält: [Auto]: Fiat Mopster released Urs erhält: [Auto]: Fiat Mopster released Was wollen Sie? (0) Programm beenden (1) neuen Kunden erstellen (2) neuen Nachrichtenkanal erstellen (3) Kunde abonniert Nachrichtenkanal (4) Nachrichtenkanalstatistik (5) Nachricht eines Nachrichtenkanals verändern 4 (1) Auto Ute zahlt 3 Urs zahlt 0 </pre>
---	---

### Lösung:

Siehe Anhang.

```
1 public class Abonnement {
2     private NachrichtenkanalObserver observer;
3     private Kunde kunde;
4     private IAbrechnungsart abrechnungsart = new PostPaid(this);
5     private int bezahlt = 0;
6
7     public Abonnement(Nachrichtenkanal kanal, Kunde kunde, int intervall) {
8         this.observer = new NachrichtenkanalObserver(this, kanal);
9         this.kunde = kunde;
10
11         assert abrechnungsart instanceof PostPaid;
12         ((PostPaid) abrechnungsart).setIntervall(intervall);
13
14         System.out.println("Abonnement erstellt");
15     }
16
17     public NachrichtenkanalObserver getObserver() {
18         return observer;
19     }
20
21     public Kunde getKunde() {
22         return kunde;
23     }
24
25     public void addBezahlt(int bezahlt) {
26         this.bezahlt += bezahlt;
27     }
28
29     @Override
30     public String toString() {
31         return "Abonnement{" +
32             "kunde=" + kunde.getName() +
33             ", bezahlt=" + bezahlt +
34             '}';
35     }
36
37     public void update() {
38         System.out.println("Abonnement: " + this.kunde.getName()
39             + " hat eine Nachricht erhalten");
40         abrechnungsart.bezahlen();
41         this.kunde.benachrichtige(
42             this.observer.getKanal().getBezeichnung() + ": "
43             + this.observer.getKanal().getInhalt());
44     }
45 }
```

Listing 1: Abonnement.java

```
1 public interface IAbrechnungsart {
2     public void bezahlen();
3 }
```

Listing 2: IAbrechnungsart.java

```
1 public class PostPaid implements IAbrechnungsart {
2     private Abonnement abo;
3     private int gesendet = 0;
4     private int intervall = 3; // default
5
6     public PostPaid(Abonnement abo) {
7         this.abo = abo;
8     }
9
10    public void setIntervall(int intervall) {
11        this.intervall = intervall;
12    }
13
14    public void bezahlen() {
15        gesendet = (gesendet + 1) % intervall;
16        if (gesendet == 0) {
17            int preis = abo.getObserver().getKanal().getPreis();
18            String name = abo.getKunde().getName();
19            System.out.println(
20                "PostPaid: " + name + ": " + preis * intervall + " Euro");
21            if (this.abo.getKunde().bezahlen(preis * intervall)) {
22                this.abo.addBezahlt(preis * intervall);
23                System.out.println("PostPaid: " + name + ": Beahlt!");
24            } else {
25                System.out.println("PostPaid: " + name + ": Konnte nicht zahlen!");
26            }
27        }
28        // sorry leude aber ich hab keine lust mehr
29        this.abo.getKunde().removeAbonnement(this.abo);
30        this.abo.getObserver().getKanal()
31            .removeObserver(this.abo.getObserver());
32        System.out.println("PostPaid: " + name + ": Abonnement beendet!");
33    }
34
35    ;
36 }
```

Listing 3: PostPaid.java

```
1  import java.util.*;
2
3  public class Kunde {
4      private String name;
5      private int kapital;
6      private List<Abonnement> abonnements = new ArrayList<>();
7
8      public Kunde(String name, int kapital) {
9          this.name = name;
10         this.kapital = kapital;
11         System.out.println("Kunde " + name + " mit " + kapital + " Euro erzeugt."
12     );
13     }
14
15     public String getName() {
16         return name;
17     }
18
19     public int getKapital() {
20         return kapital;
21     }
22
23     public void addAbonnement(Abonnement abo) {
24         abonnements.add(abo);
25     }
26
27     public void removeAbonnement(Abonnement abo) {
28         abonnements.remove(abo);
29     }
30
31     public boolean bezahlen(int betrag) {
32         if (kapital >= betrag) {
33             kapital -= betrag;
34             return true;
35         }
36         return false;
37     }
38
39     public void benachrichtige(String text) {
40         System.out.println(text);
41     }
42
43     @Override
44     public String toString() {
45         return "Kunde{" +
46             "name='" + name + '\'' +
47             ", kapital=" + kapital +
48             '}';
49     }
50 }
```

Listing 4: Kunde.java

```
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Scanner;
4
5  public class Main {
6      private static final int MAX_CAPITAL = 10000;
7      private static final int MAX_COST = 100;
8
9      private List<Kunde> kunden = new ArrayList<>();
10     private List<Nachrichtenkanal> kanaele = new ArrayList<>();
11
12     public void dialog() {
13         int eingabe = -1;
14         while (eingabe != 0) {
15             System.out.println("Was wollen Sie?\n"
16                 + " (0) Programm beenden\n"
17                 + " (1) neuen Kunden erstellen\n"
18                 + " (2) neuen Nachrichtenkanal erstellen\n"
19                 + " (3) Kunde abonniert Nachrichtenkanal\n"
20                 + " (4) Nachrichtenkanalstatistik\n"
21                 + " (5) Nachricht eines Nachrichtenkanals veraendern");
22             eingabe = new Scanner(System.in).nextInt();
23             switch (eingabe) {
24                 case 1:
25                     neuerKunde();
26                     break;
27                 case 2:
28                     neuerNachrichtenkanal();
29                     break;
30                 case 3:
31                     neuesAbonnement();
32                     break;
33                 case 4:
34                     nachrichtenkanalstatistik();
35                     break;
36                 case 5:
37                     nachrichtVeraendern();
38                     break;
39             }
40         }
41     }
42
43     private String textEingeben(String aufforderung) {
44         Scanner sc = new Scanner(System.in);
45         System.out.print(aufforderung);
46         return sc.nextLine();
47     }
48
49     // Koennt Java optionale Parameter, waere die Sprache nicht so ein
50     // Scheiss wie ohnehin schon.
51     private int nummerwaehlen(int size) {
52         return nummerwaehlen(size, null);
53     }
```

```
54
55 // (--)
56 private int nummerwaehlen(int size, String aufforderung) {
57     int eingabe = -1;
58     aufforderung = aufforderung == null ? "" : (" (" + aufforderung + ")");
59     System.out.println("Bitte geben Sie eine Nummer zwischen 1 und "
60         + size + " ein." + aufforderung);
61     while (eingabe < 1 || eingabe > size) {
62         eingabe = new Scanner(System.in).nextInt();
63     }
64     return eingabe;
65 }
66
67 private Nachrichtenkanal nachrichtenkanalWaehlen() {
68     int i = 1;
69     System.out.println("Welcher Nachrichtenkanal? ");
70     for (Nachrichtenkanal n : kanaele)
71         System.out.println(" (" + (i++) + ") " + n);
72     return kanaele.get(nummerwaehlen(kanaele.size() - 1));
73 }
74
75 private void nachrichtVeraendern() {
76     Nachrichtenkanal nk = nachrichtenkanalWaehlen();
77     nk.updateInhalt(textEingeben("Neuer Inhalt: "));
78 }
79
80 private void nachrichtenkanalstatistik() {
81     int i = 1;
82     for (Nachrichtenkanal n : kanaele) {
83         System.out.println(" (" + (i++) + ") " + n);
84         for (NachrichtenkanalObserver observer : n.getObservers())
85             System.out.println("\t" + observer.getAbo());
86     }
87 }
88
89 private void neuesAbonnement() {
90     if (kunden.isEmpty()) {
91         System.out.println("Es gibt noch keine Kunden.");
92         return;
93     }
94     if (kanaele.isEmpty()) {
95         System.out.println("Es gibt noch keine Nachrichtenkanale.");
96         return;
97     }
98     int i = 1;
99     System.out.println("Welcher Kunde? ");
100     for (Kunde k : kunden)
101         System.out.println(" (" + (i++) + ") " + k);
102     Kunde kunde = kunden.get(nummerwaehlen(kunden.size() - 1));
103     Nachrichtenkanal nk = nachrichtenkanalWaehlen();
104     kunde.addAbonnement(new Abonnement(nk, kunde,
105         nummerwaehlen(10, "Zahlungsintervall")));
106 }
107
108 private void neuerNachrichtenkanal() {
```

```
109         kanaele.add(new Nachrichtenkanal(  
110             textEingeben("Bezeichnung des Nachrichtenkanals: "),  
111             textEingeben("Inhalt des Nachrichtenkanals: "),  
112             nummerwaehlen(MAX_COST, "Preis")));  
113     }  
114  
115     private void neuerKunde() {  
116         kunden.add(new Kunde(  
117             textEingeben("Name des Kunden: "),  
118             nummerwaehlen(MAX_CAPITAL, "Kapital")));  
119     }  
120  
121     public static void main(String[] args) {  
122         new Main().dialog();  
123     }  
124  
125 }
```

Listing 5: Main.java



```
1  import java.util.*;
2
3  public class Nachrichtenkanal {
4      private String bezeichnung;
5      private String inhalt;
6      private int preis;
7      private ArrayList<NachrichtenkanalObserver> observers = new ArrayList<>();
8
9      public Nachrichtenkanal(String bezeichnung, String inhalt, int preis) {
10         this.bezeichnung = bezeichnung;
11         this.inhalt = inhalt;
12         this.preis = preis;
13         System.out.println("Nachrichtenkanal " + bezeichnung
14             + " mit Preis von " + preis + " Euro erzeugt.");
15     }
16
17     public String getBezeichnung() {
18         return bezeichnung;
19     }
20
21     public String getInhalt() {
22         return inhalt;
23     }
24
25     public int getPreis() {
26         return preis;
27     }
28
29     public ArrayList<NachrichtenkanalObserver> getObservers() {
30         return observers;
31     }
32
33     public void addObserver(NachrichtenkanalObserver observer) {
34         observers.add(observer);
35     }
36
37     public void removeObserver(NachrichtenkanalObserver observer) {
38         observers.remove(observer);
39     }
40
41     @Override
42     public String toString() {
43         return "Nachrichtenkanal{" +
44             "bezeichnung='" + bezeichnung + '\'' +
45             ", inhalt='" + inhalt + '\'' +
46             ", preis='" + preis + '\'' +
47             ", abonnenten='" + observers.size() + '\'' +
48             '}';
49     }
50
51     public void updateInhalt(String neuerInhalt) {
52         this.inhalt = neuerInhalt;
53         this.update();
54     }
```

```
55
56     public void update() {
57         // temporaere Variable damit der Code in die Zeile passt
58         var temp = (ArrayList<NachrichtenkanalObserver>) this.observers.clone();
59         for (NachrichtenkanalObserver observer : temp) {
60             observer.update();
61         }
62     }
63 }
```

Listing 6: Nachrichtenkanal.java

```
1 public class NachrichtenkanalObserver {
2     private Abonnement abo;
3     private Nachrichtenkanal kanal;
4
5     public NachrichtenkanalObserver(Abonnement abo, Nachrichtenkanal kanal) {
6         this.abo = abo;
7         this.kanal = kanal;
8         kanal.addObserver(this);
9         System.out.println("NachrichtenkanalObserver erstellt");
10    }
11
12    public Abonnement getAbo() {
13        return abo;
14    }
15
16    public void setAbo(Abonnement abo) {
17        this.abo = abo;
18    }
19
20    public Nachrichtenkanal getKanal() {
21        return kanal;
22    }
23
24    public void setKanal(Nachrichtenkanal kanal) {
25        this.kanal = kanal;
26    }
27
28    public void update() {
29        this.abo.update();
30    }
31 }
```

Listing 7: NachrichtenkanalObserver.java