

Softwaretechnik

Hausaufgabenblatt 10

Patrick Gustav Blaneck

Letzte Änderung: 17. Januar 2022

1. Politiker

Mit dem folgenden Zustandsdiagramm wird das Verhalten eines fiktiven Politikers beschrieben. Beachten Sie, dass es eine Parallelkomposition und mehrere hierarchische Zustände gibt.

Eine Kantenbeschriftung e/a bedeutet, dass nach Erhalt des Ereignisses e eine Aktion a (hier ein internes Ereignis) ausgelöst wird. Weiterhin stellen wir Ihnen ein Programmfragment zur Verfügung, mit dem der Nutzer ein solches Objekt steuern kann. Die Grundidee ist dabei, dass der Nutzer eine Eingabe macht und das Objekt genau seinen aktuellen Zustand ausgibt. Der bereits implementierte Dialog in der hier zur Verfügung gestellten Klasse `Steuerung`, die noch ergänzt werden muss, hat folgendes Aussehen:

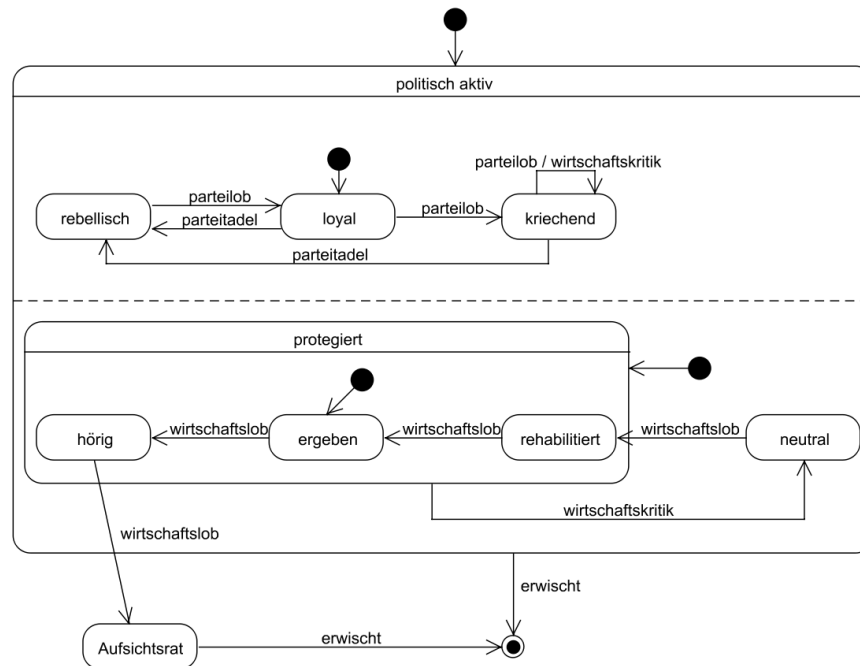
```
1 Welches naechste Ereignis?
2 (0) Lob von der eigenen Partei
3 (1) Tadel von der eigenen Partei
4 (2) Lob von der Wirtschaft
5 (3) Erwischt
6 1
7 Fiktiver Politiker befindet sich in (Teil)-Zustaenden:
8 POLITISCH_AKTIV REBELLISCH PROTEGIERT ERGEBEN
```

Man sieht die Ausgabe, nachdem der Nutzer das Ereignis `parteitadel` ausgelöst hat und die genauen Informationen über den aktuellen Zustand ausgegeben werden.

```
1  import java.util.Scanner;
2
3  public class Steuerung {
4      private Context_Politiker politiker;
5
6      public Steuerung(){
7          int eingabe=-1;
8          politiker=new Context_Politiker();
9          while(politiker.aktiv()){
10             System.out.println("Welches naechste Ereignis?\n"
11                 +" (0) Lob von der eigenen Partei\n"
12                 +" (1) Tadel von der eigenen Partei\n"
13                 +" (2) Lob von der Wirtschaft\n"
14                 +" (3) Erwischt");
15             eingabe=new Scanner(System.in).nextInt();
16             switch(eingabe){
17                 case 0:
18                     politiker.parteilob();
19                     break;
20                 case 1:
21                     politiker.parteitadel();
22                     break;
23                 case 2:
24                     politiker.wirtschaftslob();
25                     break;
26                 case 3:
27                     politiker.erwischt();
28                     break;
29             }
30             System.out.println("Fiktiver Politiker befindet sich in"
31                 +" (Teil)-Zustaenden: " + politiker);
32         }
33     }
34
35     public static void main(String[] args) {
36         new Steuerung();
37     }
38 }
39
```

Listing 1: Steuerung.java

Dieser Zustandsautomat zeigt alle möglichen Zustände und Unterzustände eines Politikers sowie deren Übergänge.



Hinweis: Dem Endzustand können Sie z.B. den Namen Ruhestand oder Ende geben (in der Implementierung und ggf. bei Aufgabenteil (a)).

(a) Welchen Zustand erreicht der Automat, nachdem die folgenden Ereignisse aufgetreten sind? (getrennt betrachten)

i. `parteilob.parteitadel`

Lösung:

POLITISCH_AKTIV REBELLISCH PROTEGIERT ERGEBEN

ii. `parteilob.wirtschaftslob.wirtschaftslob`

Lösung:

POLITISCH_AKTIV KRIECHEND AUFSICHTSRAT

iii. `parteilob.wirtschaftslob.parteilob.wirtschaftslob`

Lösung:

POLITISCH_AKTIV KRIECHEND PROTEGIERT REHABILITIERT

iv. `parteilob.wirtschaftslob.erwischt`

Lösung:

RUHESTAND


```
1  import java.util.Scanner;
2
3  public class Steuerung {
4      private Context_Politiker politiker;
5
6      public Steuerung() {
7          int eingabe = -1;
8          politiker = new Context_Politiker();
9          while (politiker.aktiv()) {
10             System.out.println("Welches naechste Ereignis?\n"
11                 + " (0) Lob von der eigenen Partei\n"
12                 + " (1) Tadel von der eigenen Partei\n"
13                 + " (2) Lob von der Wirtschaft\n"
14                 + " (3) Erwischt");
15             eingabe = new Scanner(System.in).nextInt();
16             switch (eingabe) {
17                 case 0:
18                     politiker.parteilob();
19                     break;
20                 case 1:
21                     politiker.parteitadel();
22                     break;
23                 case 2:
24                     politiker.wirtschaftslob();
25                     break;
26                 case 3:
27                     politiker.erwischt();
28                     break;
29             }
30             System.out.println("Fiktiver Politiker befindet sich in"
31                 + " (Teil)-Zustaenden: " + politiker);
32         }
33     }
34
35     public static void main(String[] args) {
36         new Steuerung();
37     }
38 }
```

Listing 2: Steuerung.java

```
1 public class Context_Politiker {
2     // member
3     private State parteiState;
4     private State wirtschaftState;
5
6     // setter
7     public void setParteiState(State parteiState) {
8         this.parteiState = parteiState;
9     }
10
11     public void setWirtschaftState(State wirtschaftState) {
12         this.wirtschaftState = wirtschaftState;
13     }
14
15     // aktiv
16     public boolean aktiv() {
17         // parteiState = ruhestand <=> wirtschaftState = ruhestand
18         return this.parteiState != this.ruhestand;
19     }
20
21     // 'politisch aktiv' states
22     // --> 'partei' states
23     private State rebellisch;
24     private State loyal;
25     private State kriechend;
26
27     public State getRebellisch() {
28         return this.rebellisch;
29     }
30
31     public State getLoyal() {
32         return this.loyal;
33     }
34
35     public State getKriechend() {
36         return this.kriechend;
37     }
38
39     // --> 'wirtschaft' states
40     private State hoerig;
41     private State ergeben;
42     private State rehabilitiert;
43     private State neutral;
44
45     public State getHoerig() {
46         return this.hoerig;
47     }
48
49     public State getErgeben() {
50         return this.ergeben;
51     }
52
53     public State getRehabilitiert() {
54         return this.rehabilitiert;
```

```
55     }
56
57     public State getNeutral() {
58         return this.neutral;
59     }
60
61     // nicht-'politisch aktiv' states
62     private State aufsichtsrat;
63     private State ruhestand;
64
65     public State getRuhestand() {
66         return this.ruhestand;
67     }
68
69     public State getAufsichtsrat() {
70         return this.aufsichtsrat;
71     }
72
73     // ereignisse
74     public void parteilob() {
75         parteiState.parteilob();
76     }
77
78     public void parteitadel() {
79         parteiState.parteitadel();
80     }
81
82     public void wirtschaftslob() {
83         wirtschaftState.wirtschaftslob();
84     }
85
86     public void wirtschaftskritik() {
87         wirtschaftState.wirtschaftskritik();
88     }
89
90     public void erwischt() {
91         parteiState.erwischt();
92         wirtschaftState.erwischt(); // redundant
93     }
94
95     @Override
96     public String toString() {
97         // parteiState = ruhestand <=> wirtschaftState = ruhestand
98         if (parteiState == this.ruhestand)
99             return parteiState.toString();
100         return parteiState.toString() + " " + wirtschaftState.toString();
101     }
102
103     public Context_Politiker() {
104         this.rebellisch = new Rebellisch(this);
105         this.loyal = new Loyal(this);
106         this.kriechend = new Kriechend(this);
107
108         this.hoerig = new Hoerig(this);
109         this.ergeben = new Ergeben(this);
```

```
110     this.rehabilitiert = new Rehabilitiert(this);  
111     this.neutral = new Neutral(this);  
112  
113     this.aufsichtsrat = new Aufsichtsrat(this);  
114     this.ruhestand = new Ruhestand(this);  
115  
116     this.parteiState = this.loyal;  
117     this.wirtschaftState = this.ergeben;  
118 }  
119 }
```

Listing 3: Context_Politiker.java


```
1 public interface State {  
2     public void parteilob();  
3  
4     public void parteitadel();  
5  
6     public void wirtschaftslob();  
7  
8     public void wirtschaftskritik();  
9  
10    public void erwischt();  
11 }
```

Listing 4: State.java

```
1 class Aufsichtsrat implements State {
2     private final Context_Politiker context;
3
4     public Aufsichtsrat(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "AUFSICHTSRAT";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20     }
21
22     public void wirtschaftskritik() {
23     }
24
25     public void erwischt() {
26         context.setParteiState(context.getRuhestand());
27         context.setWirtschaftState(context.getRuhestand());
28     }
29
30 }
```

Listing 5: Aufsichtsrat.java

```
1  class Ergeben implements State {
2      private final Context_Politiker context;
3
4      public Ergeben(Context_Politiker context) {
5          this.context = context;
6      }
7
8      @Override
9      public String toString() {
10         return "PROTEGIERT ERGEBEN";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20         context.setWirtschaftState(context.getHoerig());
21     }
22
23     public void wirtschaftskritik() {
24         context.setWirtschaftState(context.getNeutral());
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32 }
```

Listing 6: Ergeben.java

```
1 class Hoerig implements State {
2     private final Context_Politiker context;
3
4     public Hoerig(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "PROTEGIERT HOERIG";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20         context.setWirtschaftState(context.getAufsichtsrat());
21     }
22
23     public void wirtschaftskritik() {
24         context.setWirtschaftState(context.getNeutral());
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32 }
```

Listing 7: Hoerig.java

```
1 class Kriechend implements State {
2     private final Context_Politiker context;
3
4     public Kriechend(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "POLITISCH_AKTIV KRIECHEND";
11     }
12
13     public void parteilob() {
14         context.wirtschaftskritik();
15     }
16
17     public void parteitadel() {
18         context.setParteiState(context.getRebellisch());
19     }
20
21     public void wirtschaftslob() {
22     }
23
24     public void wirtschaftskritik() {
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32 }
```

Listing 8: Kriechend.java

```
1  class Loyal implements State {
2      private final Context_Politiker context;
3
4      public Loyal(Context_Politiker context) {
5          this.context = context;
6      }
7
8      @Override
9      public String toString() {
10         return "POLITISCH_AKTIV LOYAL";
11     }
12
13     public void parteilob() {
14         context.setParteiState(context.getKriechend());
15     }
16
17     public void parteitadel() {
18         context.setParteiState(context.getRebellisch());
19     }
20
21     public void wirtschaftslob() {
22     }
23
24     public void wirtschaftskritik() {
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32
```

Listing 9: Loyal.java

```
1  class Neutral implements State {
2      private final Context_Politiker context;
3
4      public Neutral(Context_Politiker context) {
5          this.context = context;
6      }
7
8      @Override
9      public String toString() {
10         return "NEUTRAL";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20         context.setWirtschaftState(context.getRehabilitiert());
21     }
22
23     public void wirtschaftskritik() {
24         // context.setWirtschaftState(context.getKommunistisch());
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32 }
```

Listing 10: Neutral.java

```
1 class Rebellisch implements State {
2     private final Context_Politiker context;
3
4     public Rebellisch(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "POLITISCH_AKTIV REBELLISCH";
11     }
12
13     public void parteilob() {
14         context.setParteiState(context.getLoyal());
15     }
16
17     public void parteitadel() {
18     }
19
20     public void wirtschaftslob() {
21     }
22
23     public void wirtschaftskritik() {
24     }
25
26     public void erwischt() {
27         context.setParteiState(context.getRuhestand());
28         context.setWirtschaftState(context.getRuhestand());
29     }
30
31 }
```

Listing 11: Rebellisch.java


```
1 class Rehabilitiert implements State {
2     private final Context_Politiker context;
3
4     public Rehabilitiert(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "PROTEGIERT REHABILITIERT";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20         context.setWirtschaftState(context.getErgeben());
21     }
22
23     public void wirtschaftskritik() {
24         context.setWirtschaftState(context.getNeutral());
25     }
26
27     public void erwischt() {
28         context.setParteiState(context.getRuhestand());
29         context.setWirtschaftState(context.getRuhestand());
30     }
31 }
32 }
```

Listing 12: Rehabilitiert.java

```
1 class Ruhestand implements State {
2     private final Context_Politiker context;
3
4     public Ruhestand(Context_Politiker context) {
5         this.context = context;
6     }
7
8     @Override
9     public String toString() {
10         return "RUHESTAND";
11     }
12
13     public void parteilob() {
14     }
15
16     public void parteitadel() {
17     }
18
19     public void wirtschaftslob() {
20     }
21
22     public void wirtschaftskritik() {
23     }
24
25     public void erwischt() {
26     }
27
28 }
```

Listing 13: Ruhestand.java