

Softwaretechnik

Hausaufgabenblatt 10

Patrick Gustav Blaneck

Letzte Änderung: 11. Dezember 2021

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:

```
Welches naechste Ereignis?  
(0) Lob von der eigenen Partei  
(1) Tadel von der eigenen Partei  
(2) Lob von der Wirtschaft  
(3) Erwischt  
1  
Fiktiver Politiker befindet sich in (Teil)-Zustaenden:  
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.

```
import java.util.Scanner;

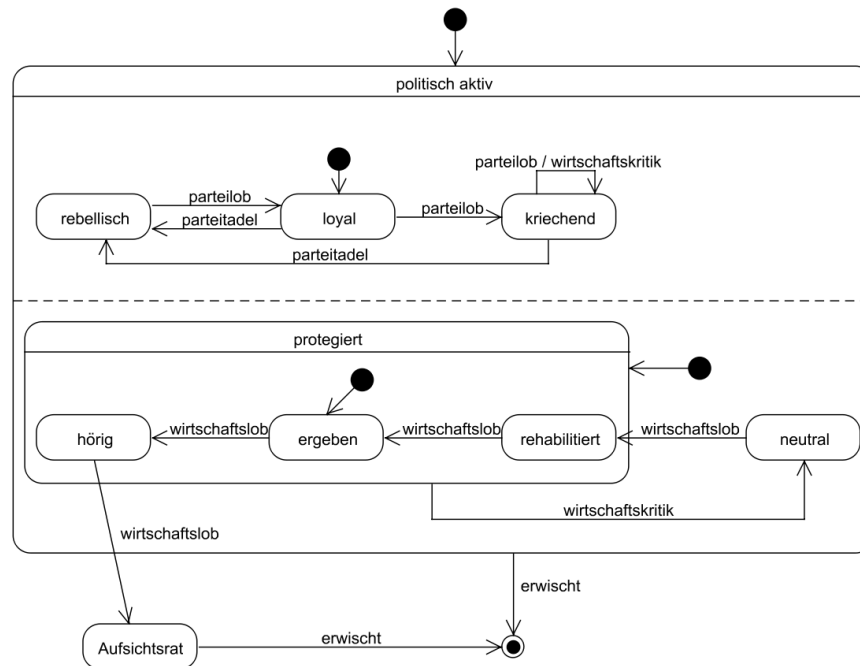
public class Steuerung {
    private Context_Politiker politiker;

    public Steuerung(){
        int eingabe=-1;
        politiker=new Context_Politiker();
        while(politiker.aktiv()){
            System.out.println("Welches naechste Ereignis?\n"
                +" (0) Lob von der eigenen Partei\n"
                +" (1) Tadel von der eigenen Partei\n"
                +" (2) Lob von der Wirtschaft\n"
                +" (3) Erwischt");
            eingabe=new Scanner(System.in).nextInt();
            switch(eingabe){
                case 0:
                    politiker.parteilob();
                    break;
                case 1:
                    politiker.parteitadel();
                    break;
                case 2:
                    politiker.wirtschaftslob();
                    break;
                case 3:
                    politiker.erwischt();
                    break;
            }
            System.out.println("Fiktiver Politiker befindet sich in"
                +" (Teil)-Zustaenden: " + politiker);
        }
    }

    public static void main(String[] args) {
        new Steuerung();
    }
}
```

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:

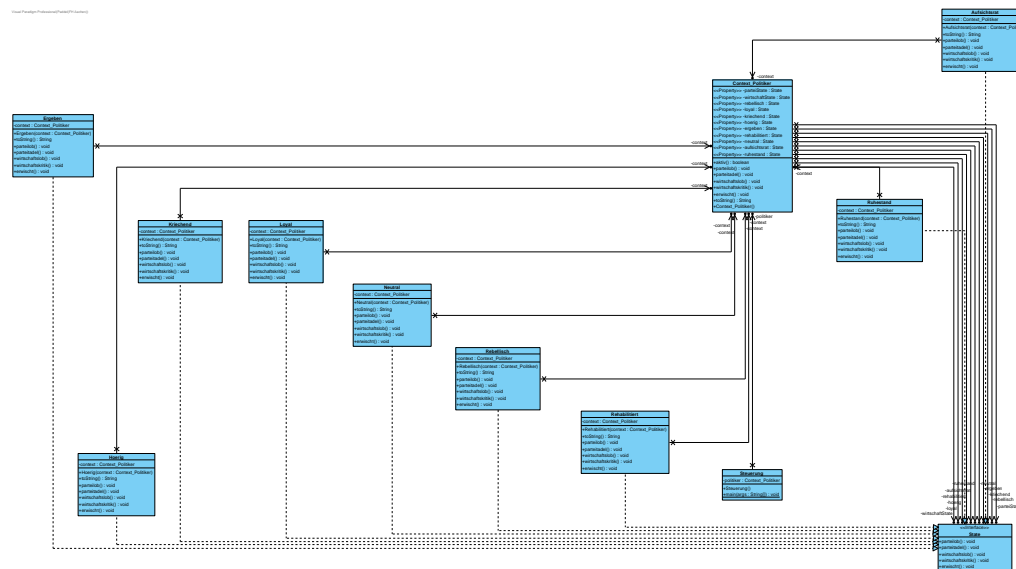
RUHESTAND

- (b) Ihre Aufgabe besteht darin, den Zustandsautomaten präzise unter Nutzung und Ergänzung der Klasse Steuerung (Quellcode siehe oben) zu *implementieren*.

Recherchieren Sie das *State-Pattern* und setzen Sie den obigen Zustandsautomaten damit um. Bedenken Sie dabei, dass ein konkreter Zustand durchaus auch der Context für weitere Unterzustände sein kann. Erzeugen Sie im Anschluss an die Implementierung mit Hilfe von Visual Paradigm ein *Klassendiagramm* aus ihrem Code (Sie können natürlich auch mit dem Klassendiagramm beginnen und daraus Code erzeugen!). Überprüfen Sie mit Ihrem Code Ihre Ergebnisse aus Aufgabenteil (a).

Weitere Informationen zur *Code Generation* mit Visual Paradigm finden Sie [hier](#).

Lösung:



Der Code befindet sich im Anhang.

Eine gute Erklärung zum State-Pattern ist [hier](#) zu finden.

P.S. Machen Sie sich Gedanken darüber, wie eine Aktion wie z.B. wirtschaftskritik aus dem Unterzustand politisch aktiv an protegiert weitergeleitet werden kann.

```
import java.util.Scanner;

public class Steuerung {
    private Context_Politiker politiker;

    public Steuerung() {
        int eingabe = -1;
        politiker = new Context_Politiker();
        while (politiker.aktiv()) {
            System.out.println("Welches naechste Ereignis?\n"
                + " (0) Lob von der eigenen Partei\n"
                + " (1) Tadel von der eigenen Partei\n"
                + " (2) Lob von der Wirtschaft\n"
                + " (3) Erwischt");
            eingabe = new Scanner(System.in).nextInt();
            switch (eingabe) {
                case 0:
                    politiker.parteilob();
                    break;
                case 1:
                    politiker.parteitadel();
                    break;
                case 2:
                    politiker.wirtschaftslob();
                    break;
                case 3:
                    politiker.erwischt();
                    break;
            }
            System.out.println("Fiktiver Politiker befindet sich in"
                + " (Teil)-Zustaenden: " + politiker);
        }
    }

    public static void main(String[] args) {
        new Steuerung();
    }
}
```

Listing 2: Steuerung.java

```
public class Context_Politiker {
    // member
    private State parteiState;
    private State wirtschaftState;

    // setter
    public void setParteiState(State parteiState) {
        this.parteiState = parteiState;
    }

    public void setWirtschaftState(State wirtschaftState) {
        this.wirtschaftState = wirtschaftState;
    }

    // aktiv
    public boolean aktiv() {
        // parteiState = ruhestand <=> wirtschaftState = ruhestand
        return this.parteiState != this.ruhestand;
    }

    // 'politisch aktiv' states
    // --> 'partei' states
    private State rebellisch;
    private State loyal;
    private State kriechend;

    public State getRebellisch() {
        return this.rebellisch;
    }

    public State getLoyal() {
        return this.loyal;
    }

    public State getKriechend() {
        return this.kriechend;
    }

    // --> 'wirtschaft' states
    private State hoerig;
    private State ergeben;
    private State rehabilitiert;
    private State neutral;

    public State getHoerig() {
        return this.hoerig;
    }

    public State getErgeben() {
        return this.ergeben;
    }

    public State getRehabilitiert() {
        return this.rehabilitiert;
    }
}
```

```
}

public State getNeutral() {
    return this.neutral;
}

// nicht-'politisch aktiv' states
private State aufsichtsrat;
private State ruhestand;

public State getRuhestand() {
    return this.ruhestand;
}

public State getAufsichtsrat() {
    return this.aufsichtsrat;
}

// ereignisse
public void parteilob() {
    parteiState.parteilob();
}

public void parteitadel() {
    parteiState.parteitadel();
}

public void wirtschaftslob() {
    wirtschaftState.wirtschaftslob();
}

public void wirtschaftskritik() {
    wirtschaftState.wirtschaftskritik();
}

public void erwischt() {
    parteiState.erwischt();
    wirtschaftState.erwischt(); // redundant
}

@Override
public String toString() {
    // parteiState = ruhestand <=> wirtschaftState = ruhestand
    if (parteiState == this.ruhestand)
        return parteiState.toString();
    return parteiState.toString() + " " + wirtschaftState.toString();
}

public Context_Politiker() {
    this.rebellisch = new Rebellisch(this);
    this.loyal = new Loyal(this);
    this.kriechend = new Kriechend(this);

    this.hoerig = new Hoerig(this);
    this.ergeben = new Ergeben(this);
}
```

```
        this.rehabilitiert = new Rehabilitiert(this);
        this.neutral = new Neutral(this);

        this.aufsichtsrat = new Aufsichtsrat(this);
        this.ruhestand = new Ruhestand(this);

        this.parteiState = this.loyal;
        this.wirtschaftState = this.ergeben;
    }
}
```

Listing 3: Context_Politiker.java


```
public interface State {  
    public void parteilob();  
  
    public void parteitadel();  
  
    public void wirtschaftslob();  
  
    public void wirtschaftskritik();  
  
    public void erwischt();  
}
```

Listing 4: State.java

```
class Aufsichtsrat implements State {
    private final Context_Politiker context;

    public Aufsichtsrat(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "AUFSICHTSRAT";
    }

    public void parteilob() {
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
    }

    public void wirtschaftskritik() {
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 5: Aufsichtsrat.java

```
class Ergeben implements State {
    private final Context_Politiker context;

    public Ergeben(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "PROTEGIERT ERGEBEN";
    }

    public void parteilob() {
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
        context.setWirtschaftState(context.getHoerig());
    }

    public void wirtschaftskritik() {
        context.setWirtschaftState(context.getNeutral());
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 6: Ergeben.java

```
class Hoerig implements State {
    private final Context_Politiker context;

    public Hoerig(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "PROTEGIERT HOERIG";
    }

    public void parteilob() {
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
        context.setWirtschaftState(context.getAufsichtsrat());
    }

    public void wirtschaftskritik() {
        context.setWirtschaftState(context.getNeutral());
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 7: Hoerig.java

```
class Kriechend implements State {
    private final Context_Politiker context;

    public Kriechend(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "POLITISCH_AKTIV KRIECHEND";
    }

    public void parteilob() {
        context.wirtschaftskritik();
    }

    public void parteitadel() {
        context.setParteiState(context.getRebellisch());
    }

    public void wirtschaftslob() {
    }

    public void wirtschaftskritik() {
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 8: Kriechend.java

```
class Loyal implements State {
    private final Context_Politiker context;

    public Loyal(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "POLITISCH_AKTIV LOYAL";
    }

    public void parteilob() {
        context.setParteiState(context.getKriechend());
    }

    public void parteitadel() {
        context.setParteiState(context.getRebellisch());
    }

    public void wirtschaftslob() {
    }

    public void wirtschaftskritik() {
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 9: Loyal.java

```
class Neutral implements State {
    private final Context_Politiker context;

    public Neutral(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "NEUTRAL";
    }

    public void parteilob() {
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
        context.setWirtschaftState(context.getRehabilitiert());
    }

    public void wirtschaftskritik() {
        // context.setWirtschaftState(context.getKommunistisch());
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 10: Neutral.java

```
class Rebellisch implements State {
    private final Context_Politiker context;

    public Rebellisch(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "POLITISCH_AKTIV REBELLISCH";
    }

    public void parteilob() {
        context.setParteiState(context.getLoyal());
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
    }

    public void wirtschaftskritik() {
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 11: Rebellisch.java


```
class Rehabilitiert implements State {
    private final Context_Politiker context;

    public Rehabilitiert(Context_Politiker context) {
        this.context = context;
    }

    @Override
    public String toString() {
        return "PROTEGIERT REHABILITIERT";
    }

    public void parteilob() {
    }

    public void parteitadel() {
    }

    public void wirtschaftslob() {
        context.setWirtschaftState(context.getErgeben());
    }

    public void wirtschaftskritik() {
        context.setWirtschaftState(context.getNeutral());
    }

    public void erwischt() {
        context.setParteiState(context.getRuhestand());
        context.setWirtschaftState(context.getRuhestand());
    }
}
```

Listing 12: Rehabilitiert.java

```
class Ruhestand implements State {  
    private final Context_Politiker context;  
  
    public Ruhestand(Context_Politiker context) {  
        this.context = context;  
    }  
  
    @Override  
    public String toString() {  
        return "RUHESTAND";  
    }  
  
    public void parteilob() {  
    }  
  
    public void parteitadel() {  
    }  
  
    public void wirtschaftslob() {  
    }  
  
    public void wirtschaftskritik() {  
    }  
  
    public void erwischt() {  
    }  
}
```

Listing 13: Ruhestand.java