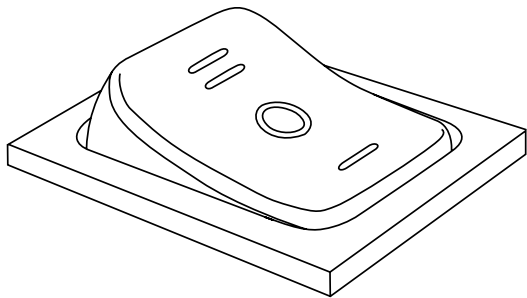


Das Zustandsmuster – Eine Einführung

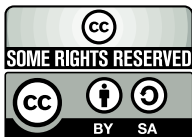


Julian Fietkau

Universität Hamburg

25. August 2010

Organisatorisches vorweg



Diese Folien sind unter CC-BY-SA 3.0 freigegeben.

Alle Illustrationen, soweit sie nicht von mir erstellt wurden, stammen aus dem OpenClipArt-Projekt.

Folien-Download und Feedback-Möglichkeit:

http://www.julian-fietkau.de/das_zustandsmuster

Übersicht

Was sind Entwurfsmuster?

Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster

Implementation des Beispiels

Weiteres Wissenswertes

Übersicht

Was sind Entwurfsmuster?

Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster

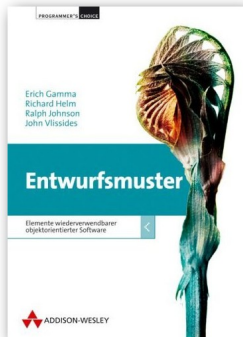
Implementation des Beispiels

Weiteres Wissenswertes

Die Grundidee (objektorientierter) Entwurfsmuster

- Beim **objektorientierten Entwurf** wird ein abstraktes Modell in Klassen bzw. Code „gegossen“. Wie genau das geschieht ist eine Frage von **Design**.
 - Hierbei tauchen bestimmte abstrakte Fragen immer wieder auf (**Meta-Probleme**).
- **Objektorientierte Entwurfsmuster** bieten schablonenartige Lösungen für solche Meta-Probleme. Es liegt am Programmierer, das Muster auf das konkrete Problem zu **übertragen**.

Standardliteratur



Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software (Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides)

Übersicht

Was sind Entwurfsmuster?

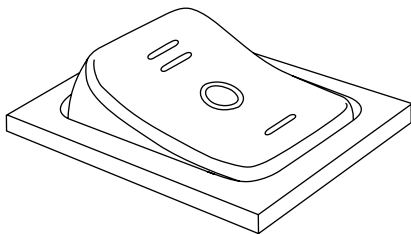
Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster

Implementation des Beispiels

Weiteres Wissenswertes

Der Kippschalter einer Lampe

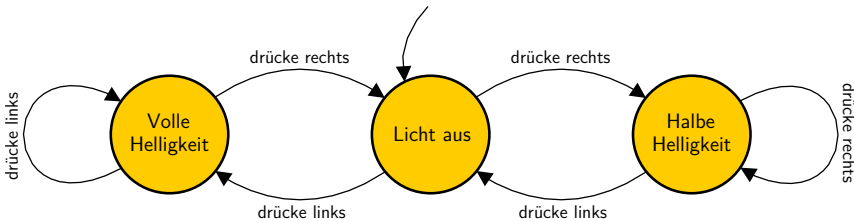


Mögliche Zustände: Licht aus, halbe Helligkeit, volle Helligkeit


Mögliche Aktionen: drücke links, drücke rechts

Zustandsabhängiges Verhalten: Wie hell ist die Lampe?

Das Zustandsdiagramm



```
1 public class Lampe {  
    private enum LampeZustand {  
        LICHT_AUS, HALBE_HELLIGKEIT, VOLLE_HELLIGKEIT;  
5    }  
  
    private LampeZustand zustand = LampeZustand.LICHT_AUS;  
  
    public void drueckeLinks() {  
10        if(zustand == LampeZustand.LICHT_AUS) {  
            zustand = LampeZustand.VOLLE_HELLIGKEIT;  
        } else if(zustand == LampeZustand.HALBE_HELLIGKEIT) {  
            zustand = LampeZustand.LICHT_AUS;  
15        }  
    }  
  
    public void drueckeRechts() {...}  
  
    public double gibHelligkeit() {  
20        double helligkeit = 0.0;  
        if(zustand == LampeZustand.LICHT_AUS) {  
            helligkeit = 0.0;  
        } else if(zustand == LampeZustand.HALBE_HELLIGKEIT) {  
            helligkeit = 0.5;  
25        } else if(zustand == LampeZustand.VOLLE_HELLIGKEIT) {  
            helligkeit = 1.0;  
        }  
        return helligkeit;  
30    }  
}
```

 Lampe.java

Übersicht

Was sind Entwurfsmuster?

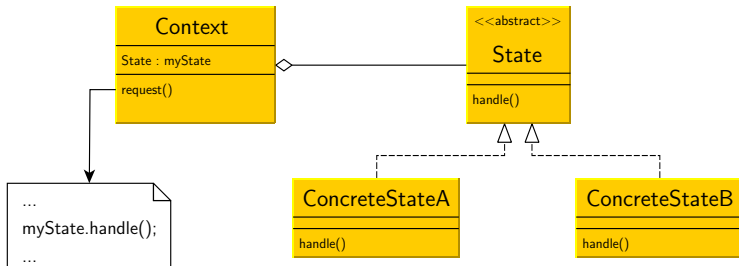
Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster

Implementation des Beispiels

Weiteres Wissenswertes

Abstrakte Darstellung des Zustandsmusters



- Der Kontext **hält** ein Zustands**objekt**
- Zustandsabhängiges Verhalten wird dorthin **delegiert**
- Wie funktionieren Übergänge?

Übersicht

Was sind Entwurfsmuster?

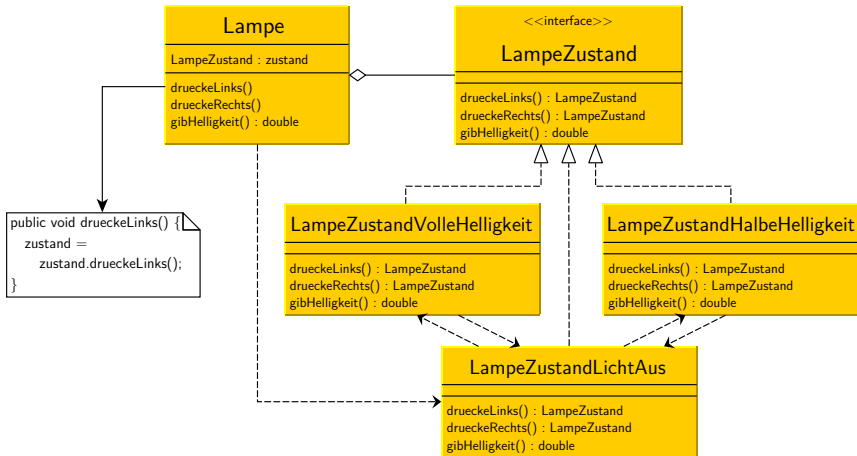
Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster


Implementation des Beispiels

Weiteres Wissenswertes


Eine Variante: dezentrale Übergänge




```
1 public class Lampe {  
    private LampeZustand zustand;  
5 public Lampe() {  
    zustand = new LampeZustandLichtAus();  
    }  
    public void drueckeLinks() {  
10    zustand = zustand.drueckeLinks();  
    }  
    public void drueckeRechts() {  
15    zustand = zustand.drueckeRechts();  
    }  
    public double gibHelligkeit() {  
    return zustand.gibHelligkeit();  
20 }  
}
```

 Lampe.java

```
1 public interface LampeZustand {  
    public LampeZustand drueckeLinks();  
  
    public LampeZustand drueckeRechts();  
  
5     public double gibHelligkeit();  
}
```

 LampeZustand.java

```
1 public class LampeZustandHalbeHelligkeit implements LampeZustand {  
  
    public LampeZustand drueckeLinks() {  
6         return new LampeZustandLichtAus();  
    }  
  
    public LampeZustand drueckeRechts() {  
10        return this;  
    }  
  
    public double gibHelligkeit() {  
        return 0.5;  
    }  
}
```

 LampeZustandHalbeHelligkeit.java

Beobachtungen

- Das zustandsabhängige Verhalten ist besser gekapselt und vom Kontext getrennt
- Weniger Sprünge im Code, bessere Übersicht
- Aber: struktureller Overhead

Übersicht

Was sind Entwurfsmuster?

Ein Beispiel für zustandsbasiertes Verhalten

Das Zustandsmuster

Implementation des Beispiels

Weiteres Wissenswertes

Hinweise

- Es gibt **viele Varianten** der Implementierung, die unter verschiedenen Umständen sinnvoll sind.
- In aktuellen Java-Versionen erleichtern **Enums** die Umsetzung.
- Nicht jedes zustandsabhängige Verhalten muss mit dem Zustandsmuster beantwortet werden - **keep it simple**.

Modellierung zustandsorientierter Systeme in Java: Das Zustandsmuster, Varianten und Alternativen

(Julian Fietkau, Janina Nemeč)

[http://www.julian-fietkau.de/
modellierung_zustandsorientierter_systeme_in_java](http://www.julian-fietkau.de/modellierung_zustandsorientierter_systeme_in_java)

Danke für die Aufmerksamkeit!

