Implementierung endlicher Zustandsautomaten

Max Mustermann

Zusammenfassung
Das Abstract ist eine maximal 200 Worte lange Zusammenfassung des Inhalts der Arbeit, so dass sich der Leser vorab ein erstes Bild vom Inhalt machen kann.

Keywords
Endlicher Zustandsautomat, Design, Java

Hochschule Kaiserslautern
Corresponding author: max.mustermann@fh-kl.de

Inhaltsverzeichnis

Einleitung

Die Einleitung sollte ausreichenden Hintergrund für den Leser liefern, so dass er sich ohne großes Studium von Sekundärliteratur in das Thema hineindenken kann. Auch sollte die Aufgabenstellung bzw. Motivation für die vorliegende Arbeit dargelegt werden, sowie die Zielsetzung, die man erreichen will. Weiter wird auch das Thema von eventuell verwandten Themen abgegrenzt. Hier kann auch die Literatur [1, 2] und [3] vorgestellt werden.

Konkret sollt hier das Ziel und die Motivation des Projekts erläutert werden. Am Ende des Abschnitts steht eine kurze Übersicht über die weiteren Abschnitte.

Auch wenn die Einleitung zu Beginn der Arbeit liegt, wird sie oft erst am Ende verfasst.

1. Implementierungsvarianten

Kurze Einführung in den Abschnitt. Es wird beschrieben was jetzt kommt.

1.1 Prozedurale Implementierung

Beschreibung des Ansatzes `switch-case`

1.2 Objektorientierte Implementierung

Beschreibung des Ansatzes mit State-Pattern (Klassendiagramm: State, Transitions, Context)

1.3 Evaluierung der Implementierungsansätze

Beschreibung der Vor- und Nachteile der beiden Ansätze.

2. Anwendungsbeispiele

Kurze Einführung in den Abschnitt. Es wird beschrieben was jetzt kommt.

2.1 Endlicher Automat für reguläre Ausdrücke

Zielsetzung: Prüfung von Strings auf bestimmtes Format, vorgegeben durch regulären Ausdruck der durch einen Automaten realisiert ist.

2.1.1 XML-Konfiguration

Deklarative Beschreibung des Automaten

2.1.2 FSM-Framework

Frameworkansatz

2.2 Steuerung einer Digitaluhr

Erweiterte Notation mit statecharts. Komplexerer Automat mit Unterzuständenünd ”Parallelzuständen”. Implementierungsansatz.

3. Zusammenfassung

Hier wird nochmal der Inhalt und die Ergebnisse der Arbeit erörtert. Im Ausblick werden Themen und Aufgabenstellungen genannt, die es lohnt weiter zu untersuchen.

Literatur

[1] David Harel. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program.*, 8(3):231–274, June 1987.

[2] D. Harel, Y. Feldman, and M. Krieger. *Algorithmik*. Springer Berlin Heidelberg, 2006.

[3] Jilles Van Gurp and Jan Bosch. On the implementation of finite state machines. In *in Proceedings of the 3rd Annual IASTED International Conference Software Engineering and Applications, IASTED/Acta*, pages 172–178. Press, 1999.

[4] Peter Rechenberg. *Technisches Schreiben (nicht nur) für Informatiker*. Hanser-Verlag, 2002.

Erklärung zur Ausarbeitung

Hiermit erkläre ich, *Vorname Nachname (Matrikel)*, dass ich die vorliegende Ausarbeitung selbstständig und ohne fremde Hilfe angefertigt habe und keine anderen als in der Abhandlung angegebenen Hilfen benutzt habe; dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

_____

Unterschrift

1. Bemerkungen zur Ausarbeitung

Der Umfang der Ausarbeitung sollte 10 Seiten nicht übersteigen. Die Ausarbeitung sollte dem Charakter nach eher einem *scientific paper* entsprechen. Siehe hierzu auch [4]. Keine Ich-Form benutzen. Aussagen sollten möglichst belegt oder begründet werden.

Die schriftliche Ausarbeitung hat folgende Zwecke:

- die Ausarbeitung soll Kommilitonen (die nicht an der Veranstaltung teilgenommen haben) in das Thema einführen.
- die Autorin bzw. der Autor soll üben, wie man technische Sachverhalte kurz und klar beschriebt.
- die Ausarbeitung bildet die Grundlage der Bewertung. In der Arbeit soll gezeigt werden, dass man das Thema geistig durchdrungen hat.

A.1 Latex-Syntax

Im folgenden finden Sie einige nützliche Latex-Anweisungen.

Abbildung 2 zeigt ein Bild, dass die komplette Seitenbreite einnimmt. Abbildung 1 ist dagegen in die Spalte eingebettet.

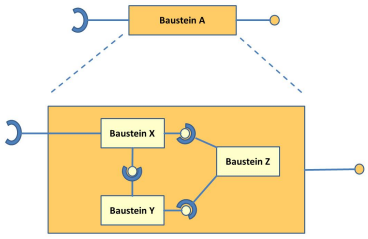

Abbildung 1. Beispiel für ein Bild in der Spalte

Hier sehen Sie, wie man ein Java-Code-Listing mit einbinden kann. Es kann auch ein Label vergeben werden, auf das dann referenziert werden kann (vgl. Listing 1).

```
public class Test
{
  private int mCount = 0;

  @Override
  public void xyz()
  {
    for(int i = 0; i< 10; i++)
    {
      this.mCount += i;
    }

    // Ein Kommentar
    this.mEnd = this.mCount;
  }
}
```

Listing 1. Eine Testklasse

Hier ein Beispiel für eine mathematische Formel ohne Nummer

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta$$

Oder mehrere mathematische Ausdrücke untereinander

$$f(x) = \frac{3e^x}{1-x^2} \quad (1)$$

$$g(x) = \sqrt[3]{\sin\alpha} \quad (2)$$

$$h(x) = \frac{2+3i}{1-i} \quad (3)$$

Hier noch ein Beispiel für eine Aufzählung, wobei die Aufzählungspunkte direkt nacheinander, d.h. ohne Leerzeile, aufgelistet werden.

1. First item in a list
2. Second item in a list
3. Third item in a list

Ein Beispiel für eine Tabelle, falls man ein solches Konstrukt benötigt.

Tabelle 1. Table of Grades

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

Siehe Tabelle 1. Und hier nochmal das Einbinden eines Bildes (vgl. Abb. 3.), wobei das Bild innerhalb der Spalte gezeigt wird.

A.2 Allgemeine Hinweise

Bei den Ausführungen fasst man sich so kurz wie möglich, aber so lange wie nötig um verständlich zu sein. Man schreibt die Projektarbeit für Leser, die die gleichen Vorkenntnisse haben wie man selbst. Die Projektarbeit besitzt im Prinzip den selben Aufbau wie dieses Dokument.

Eine Arbeit wird nur einmal geschrieben aber von vielen, d.h. oft gelesen. Der Verfasser sollte es sich deshalb bei der Formulierung schwer machen, damit es der Leser später um so leichter hat. Benutzen Sie Bilder und legen Sie einen roten Faden durch Ihren Text.

A.3 Bemerkungen zur Literatur

Die Literaturliste muss die Referenzen enthalten, auf die man vorher verwiesen hat - nicht mehr und nicht weniger. Sie werden sicherlich sehr oft referenzieren. Wenn ein Artikel eine Web-Adresse hat, muss die Web-Adresse aufgenommen werden, vgl. [**?**] oder [**?**].

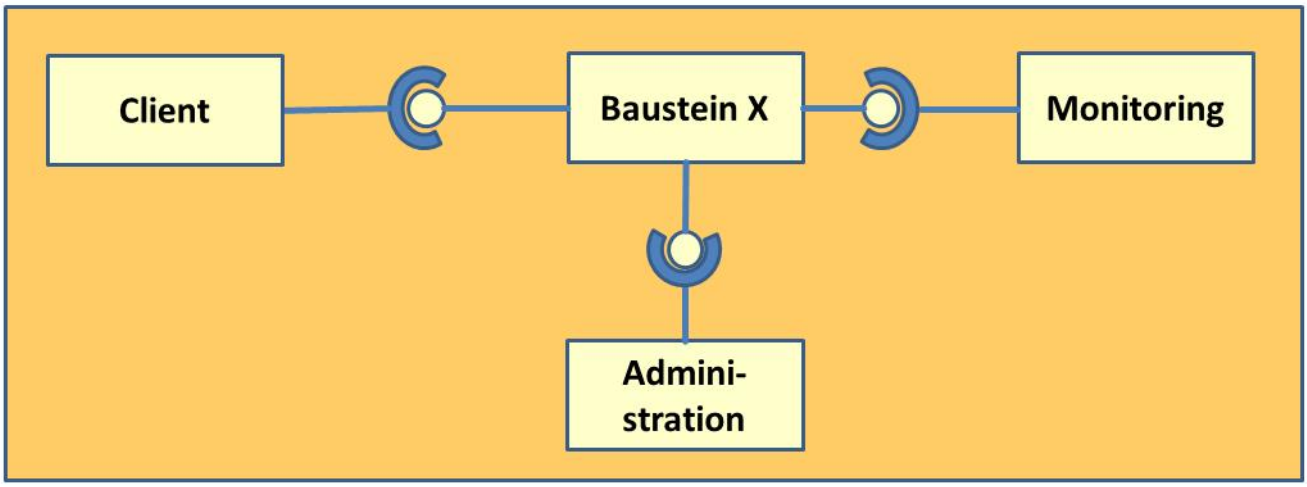

Abbildung 2. Beispiel für ein breites Bild

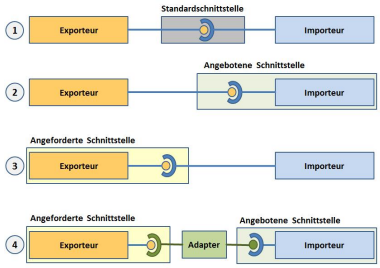

Abbildung 3. Beispiel für ein Bild in der Spalte

2. Formalien

B.1 Bewertungskriterien

Damit die Arbeit nicht nur eine reine Zusammenfassung des Vorlesungsinhalts bleibt, soll die Projektarbeit auch einen *Eigenanteil* enthalten. Dieser Eigenanteil kann z.B. eine eigene Literaturrecherche sein und somit Themen aufgreifen, die nicht explizit in der Veranstaltung besprochen wurden.

In die Bewertung der Projektarbeit gehen folgende Punkte mit ein:

1. Formale Kriterien: Äußere Form, Layout, Zitiertechnik und korrekte Angabe der Literatur, Stil, Abbildungen, Rechtschreibung, etc.
2. Systematik der Darstellung: Vollständigkeit (Wie breit und tiefgehend wird das Thema behandelt?), korrekte Wiedergabe (wurde das Thema richtig verstanden), logische Gliederung und Gedankenführung
3. Eigenständigkeit der Darstellung: Wurde in eigenen Worten zusammengefasst oder nur Zitate benutzt? Präsentation des Themas (Didaktik), wurden Aussagen zueinander in Beziehung gesetzt und wurde ein eigener gedanklicher Aufbau gewählt?
4. Tiefe und Umfang des Eigenanteils: Hat der Eigenanteil fachliche Substanz? Innovationspotential, d.h. wie viele eigene Ideen wurden eingebracht.

3. Listings

Hier im Anhang sind Listings aufgeführt, die besser im „großen" einspaltigen Format wiedergegeben werden.

```
import spellchecker.algorithm.EditDistance;

public class LevenshteinAlgorithm implements EditDistance
{
  public final int distance(String from, String to)
  {
     assert from != null && to != null : "Parameters should not be null";

     // Sonderfaelle
     if (from.equals(to))    return 0;
     if (from.length() == 0) return to.length();
     if (to.length() == 0)   return from.length();

     int width = from.length() + 1;
     int height = to.length() + 1;

     // Tabelle
     int[][] table = new int[height][width];

     // Initialisierung erste Zeile
     for (int i = 0; i < width; i++)
     {
        table[0][i] = i;
     }

     // Zeilen (to)
     for (int i = 1; i < height; i++)
     {
        table[i][0] = i;
        // Spalten (from)
        for (int j = 1; j < width; j++)
        {
           int delta = 0;
           if (to.charAt(i - 1) != from.charAt(j - 1))
               delta = 1;

           table[i][j] = min(
                table[i][j-1] + 1, table[i-1][j] + 1, table[i-1][j-1] + delta );
        }
     }

     return table[height-1][width-1];
  }

  public final static int min(int a, int b, int c)
  {
     if (a < b && a < c)
         return a;
     else if (b < c)
        return b;
     else
       return c;
   }
}
```

Listing 2. Der Levenshtein Algorithmus