

Vorlesung SE WS 21/22 Übungsblatt 9

8.12.21

Felix Mayer

Aufgabe 1: Entwurf eines Zustandsdiagramms

In einem online Bestellsystem werden verschiedene Bestellung aufgenommen und bis hin zum eigentlichen Versand und der Lieferung erfasst. Für dieses Szenario soll ein Zustandsdiagramm erstellt werden.

- Der Prozess beginnt damit, dass der Kunde eine neue Bestellung bearbeitet.
- Der Kunde kann jederzeit neue Produkte hinzufügen und so die Bestellung erweitern.
- Wird die Bestellung vom Kunden abgeschickt, wird er dazu aufgefordert seine Adressdaten zu überprüfen.
- Ausgehend von der Überprüfung der Adressdaten kann er noch immer zurück zur Bestellungsbearbeitung, um weitere Änderungen vorzunehmen.
- Sobald der Kunde seine Adressdaten überprüft hat, wird der Lagerbestand überprüft zu den von ihm ausgewählten Produkten.
 - Sollte das Lager leer sein, wird der Kunde im Bestellvorgang darüber informiert, dass das Produkt nicht verfügbar ist. Anschließend wird der Bestellvorgang beendet.
 - Sollte das Produkt im Lager vorhanden sein, wird umgehend eine Bestellbestätigung erstellt.
- Wenn die Bestellbestätigung an den Kunden verschickt wurde, wird das Produkt für den Versand vorbereitet.
- Ist nun der Versand erfolgt, wird das Produkt entweder an die Rechnungsadresse geliefert oder an das Postfach des Kunden.
- Danach ist die Lieferung erfolgt und der Prozess beendet.

Überführe die verschiedenen Schritte aus der Beschreibung in ein UML Zustandsdiagramm, welche das online Bestell- und Liefersystem darstellt.

Das Zustandsdiagramm kannst du einerseits mit der Hand erstellen und abfotografieren und/oder einscannen und anschließend auf GitLab pushen oder du verwendest am Computer [Draw.io](https://draw.io).

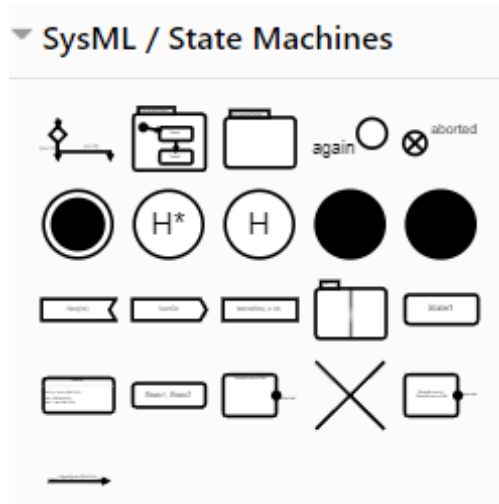
Exportiere das Diagramm und pushe es auf GitLab.

Nenne dieses „Zustandsdiagramm“ und lege es in einem Ordner „Übung9“ ab.

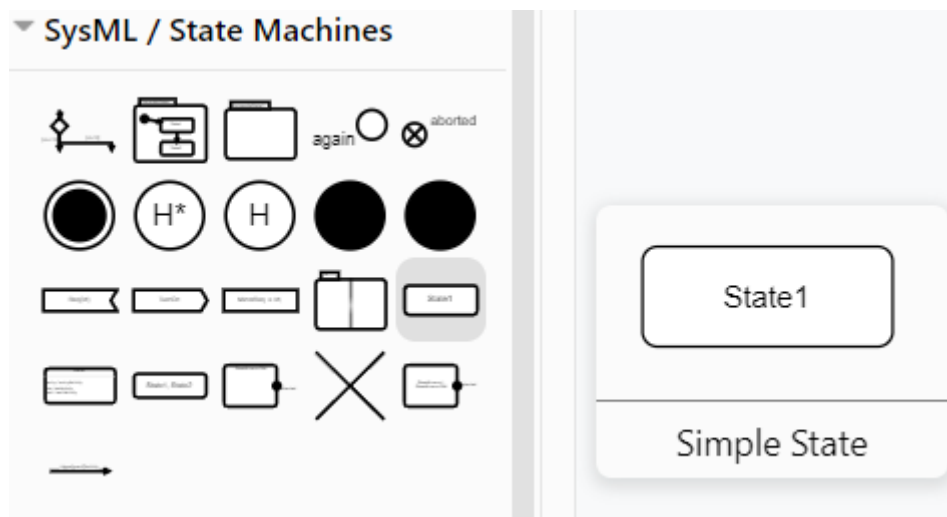
Hinweise zur Anwendung von Draw.io für Zustandsdiagramme:

Zum Einzeichnen von Zustandsdiagrammen wird ebenso die Graphik Bibliothek SysML verwendet. Die Anleitung zum Hinzufügen dieser Bibliothek liegt bereits in Aufgabenblatt 7 vor.

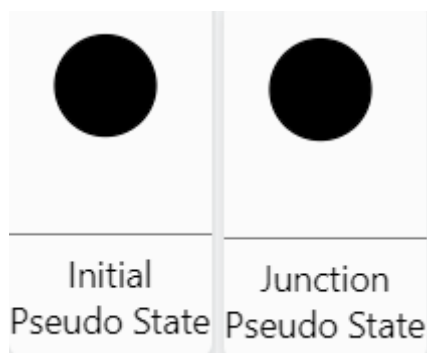
Von dieser Bibliothek wird der Abschnitt „State Machines“ benötigt.



Für die Darstellung der Zustände nutze das Graphik Element „Simple State“.



Für den Startpunkt gibt es zwei identische Graphiken. Welche davon verwendet wird, spielt keine Rolle.



Um die Zustände zu verbinden kannst du herkömmliche Pfeile aus dem Bereich „General“ verwenden.



Aufgabe 2: Implementierung im Code

Aufgabe 2.1: Entwerfen einer Zustandsvariable

Neben dem Aufgabenblatt existiert ein Src Ordner mit einer Java Datei. Diese beinhaltet die Klasse „Bestellung“.

Durch die Klasse werden Objekte instanziiert, die den Ablauf des Zustandsdiagramms widerspiegeln. Z.B. wird mit der Methode `public void überprüfeAdressdaten()` die Transition von dem Zustand, welcher der eigentlichen Bearbeitung der Bestellung dient, in den Zustand zur Überprüfung der Adressdaten dargestellt.

Zur Verwaltung der Zustände soll eine Zustandsvariable eingesetzt werden. Im Code findet sich bereits ein gleichnamiges Attribut vor, welches vom Enum Typ **Zustand** ist.

Erweitere das Enum um die notwendigen Einträge, um alle Zustände repräsentieren zu können!

Aufgabe 2.2: Vervollständigen der Transitionen

Neben der Methode **überprüfeAdressdaten** gibt es noch weitere Methoden, die die Transitionen darstellen, wie z.B. **bestellungBearbeiten**.

Vervollständige alle restlichen Methoden / Transitionen im Code. Ihr könnt euch dabei an die Vorlage von **überprüfeAdressdaten** halten.

Unter die zu vervollständigen Methoden fallen:

- bestellungBearbeiten()
- überprüfeLagerbestand()
- auswertenDesLagerbestands()
- bereiteVersandVor()
- liefereAnRechnungsadresse()
- liefereAnPostfach()

Achtet dabei auf die richtige Verwendung der Zustandsvariable.

Du kannst die korrekte Funktionsweise testen, indem du die Transitionen abspielst. Wenn du die in der Main Methode vorgegebene Reihenfolge von Transitionen ausführst, sollte das Ergebnis wie folgt aussehen:

```
public static void main(String[] args) {  
    Bestellung b = new Bestellung();  
    b.lagerbestand=10;  
  
    b.bestellungBearbeiten();  
    b.überprüfeAdressdaten();  
    b.bestellungBearbeiten();  
    b.überprüfeAdressdaten();  
    b.überprüfeLagerbestand();  
    b.auswertenDesLagerbestands();  
    b.bereiteVersandVor();  
    b.bestellungBearbeiten();  
    b.liefereAnPostfach();  
}
```

Bestellung wird bearbeitet
Adressdaten werden überprüft
Bestellung wird bearbeitet
Adressdaten werden überprüft
Lagerbestand wird überprüft
Bestätigung der Bestellung
Versand wird vorbereitet
Keine gültige Transition
Versand erfolgt an Postfach

Für die Abgabe der Aufgabe 2.1 und 2.2 pushe die erweiterte Klasse in GitLab.

Aufgabe 3: Kontrollfragen

Um welche Arte von Zustandsautomat handelt es sich? Deterministisch oder nicht deterministisch?

-

Was bedeutet es, dass eine Transition atomar ist?

-

-

Warum ist die Verwendung einer Zustandsvariable empfehlenswerter als den Zustand an den eigentlichen Objekt Attributen auszumachen?

-

-

Warum empfiehlt es sich zur Darstellung der Zustandsvariable ein Enum zu verwenden?

-

-

Wann endet der Lebenszyklus des Objekts im Hinblick auf das Diagramm und wann im Hinblick auf den Code?

-

-

Für die Abgabe der Kontrollfragen kannst du diese direkt in dem Dokument vervollständigen und es auf GitLab pushen oder du erstellst eine neue Datei dafür. Nenne diese Datei „Kontrollfragen“ und lege sie in einem Ordner „Übung9“ ab. Verwende dafür ein Dateiformat, welches unabhängig vom Betriebssystem gelesen werden kann. (.pdf, .txt, .odt, ...)