

# Fahrstuhlsteuerung

## Allgemeine Beschreibung

Die folgenden Aufgaben sind Bestandteil der Entwicklung eines Fahrstuhlsteuersystems. Als Grundannahme gehen wir dabei von einem System mit 2 Fahrstühlen aus, die jeweils 10 Stockwerke bedienen. Das Gesamtsystem teilt sich dabei in drei Komponenten:

- die Steuerung und Überwachung der Anzeigen und Kontrollen in jedem der Stockwerke
- die Steuerung und Überwachung der Anzeigen und Kontrollen in jeder Kabine
- die Umsetzung des zentralen Steuermechanismus, der die generelle Bewegung und Position der Fahrkabinen überwacht und steuert.

In jedem Stockwerk befinden sich

- je ein Knopf pro Fahrtrichtung für die Fahrstuhlanforderung
- Anzeigen für jeden Fahrstuhl über die aktuelle Position und Fahrtrichtung
- je zwei Sensoren pro Fahrstuhl, die oberhalb und unterhalb der Tür im Fahrstuhlschacht angebracht sind und bei gleichzeitiger Aktivierung anzeigen, dass die Fahrkabine in Position steht.
- eine Tür, die automatisch mit der Kabinentür geöffnet und geschlossen wird. Diese Tür besitzt keine eigene Steuerung, aber einen Sensor zur Überwachung, ob sie geöffnet oder geschlossen ist.

In jeder Fahrkabine befinden sich

- je ein Knopf pro Stockwerk zur Angabe des gewünschten Fahrziels
- eine Anzeige für die aktuelle Position und Fahrtrichtung
- ein Knopf für das Schnellschließen der Tür
- ein Knopf für das Offenhalten der Tür
- insgesamt 6 Sensoren (je drei pro Seite) in der Tür zur Erkennung von Hindernissen beim Türschließen
- ein Timer, der nach Angabe des Fahrziels oder bei vorhandenen Zieleingaben nach Öffnen der Türen als Beantwortung auf eine Anforderungen gestartet wird und bei Ablauf das Signal an die Kabinentür zum Schließen gibt; dieser Timer wird bei Betätigung des Öffne-Knopfes neu gestartet, bei Betätigung des Schließe-Knopfes gestoppt (auf 0 gesetzt).

- ein zweiter Timer, der nach Ablauf des ersten Timers aktiviert wird. Wenn nach Ablauf des zweiten Timers die Tür nicht geschlossen ist, wird ein Fehler an die zentrale Einheit gemeldet. Der zweite Timer wird zurückgesetzt, falls die Tür wieder geöffnet wird, was an einer Reaktivierung des ersten Timers erkennbar ist.

Die zentrale Steuereinheit hat folgende Aufgaben:

- Sie speichert und überwacht die aktuelle Position jeder Fahrkabine und stellt sie den Anzeigeelementen in den Stockwerken und Kabinen zur Verfügung. Als Input wird die Information der Positionssensoren in den Stockwerken verwendet.
- Sie kontrolliert Fahrrichtung und Richtungsänderungen der Kabinen; Ziel ist ein bestmögliches Scheduling in Abhängigkeit von Position und Nutzeranfragen.
- Sie überwacht mögliches Fehlverhalten des Systems, etwa Türblockierungen.

Bei der Realisierung des Systems sind eine Reihe von Regeln zu beachten, die das Gesamtverhalten bestimmen:

- Eine Fahrkabine befindet sich im Stockwerk  $n$  wenn oberer und unterer Sensor der Stockwerkserkennung im Stockwerk  $n$  aktiviert sind. Das heisst insbesondere, dass sich der Fahrstuhl solange in diesem Stockwerk befindet solange keine andere Stockwerksposition erreicht ist.
- Die Anzeigen geben die letzte gesicherte Position der Kabinen wieder (sowohl innerhalb der Kabine als auch in den Stockwerken)
- die Türen der Kabine (und damit auch die des Stockwerks) öffnen sich ausschließlich dann, wenn sich die Kabine im zugehörigen Stockwerk befindet.
- Der Fahrstuhl kann erst dann fahren, wenn innere und äußere Tür geschlossen sind.
- Das Schließen der Tür wird entweder durch den Signalgeber (Timer) nach Ablauf der Wartezeit aktiviert (Zeitsignal) oder durch den Interrupt, der vom Schnellschließen-Knopf ausgelöst wird.
- Beim Schließen der Tür geht Sicherheit vor: beim Erkennen eines Hindernisses wird ein Schnellschließen ignoriert bzw. abgebrochen.
- Das Fahrverhalten soll möglichst effizient sein; d.h. dass unnötige Richtungsänderungen vermieden werden sollen und dass auf eine Anforderung aus dem Stockwerk der jeweils nähere Fahrstuhl die Anfrage annimmt. Dabei soll die erste Bedingung eine höhere Priorität haben als die zweite.

**Anmerkung:** Die obigen Anforderungen enthalten einen minimalen Satz an Regeln. Falls bei der Bearbeitung die Notwendigkeit entsteht für ein eindeutiges Verhalten weitere Regeln aufzustellen, so sind diese in der Dokumentation entsprechend festzuhalten und zu begründen. Dabei ist zu beachten, dass keine unnötigen Einschränkungen vorgenommen werden sollen!

**Dokumentation der Aufgaben:** Neben den in Together erstellten Diagrammen sollen als Dokumentation der Lösungen der Aufgaben folgende Informationen in der Ausarbeitung zu finden sein:

- Wie wurde das Projekt in Together aufgesetzt? Welche Optionen wurden für die verschiedenen Aufgaben gewählt? Dabei soll von einer Entwicklung mit UML und Implementierung in Java ausgegangen werden.
- Alle Together-Diagramme sollen zusätzlich informell erläutert werden; dabei sollen insbesondere Designentscheidungen dokumentiert und begründet werden.

## Aufgabe 1: Use Case Diagrams

Entwickeln Sie in Together Use Case Diagramme für folgende Use Cases:

- a) Anzeigen von Informationen im Stockwerk
- b) Anforderung einer Fahrkabine in einem Stockwerk
- c) Scheduling des Fahrstuhls in der zentralen Steuerung

Welche anderen Use Cases wären im Zusammenhang mit der oben skizzierten Fahrstuhlsteuerung sinnvoll?

## Aufgabe 2: Activity Diagram

Die Aktivitäten für die Fahrstuhlanforderung vom Drücken des Knopfes bis zum Zielwahl in der Kabine als Activity-Diagramme modelliert werden.

## Aufgabe 3: Component Diagram

Beschreiben Sie die Komponenten des Gesamtsystems für die Fahrstuhlsteuerung mit ihren Schnittstellen mit Hilfe eines Component Diagrams.

Leiten Sie die Komponenten aus dem Activity-Diagramm und den allgemeinen Informationen ab.

## Aufgabe 4: Deployment Diagram

Stellen Sie in einem Deployment Diagramm die Verteilung der Fahrstuhlkomponenten und ihre Schnittstellen dar.

Wie ist die Beziehung zwischen den beiden Diagrammen?

## Aufgabe 5: State Diagrams

Entwickeln Sie Zustandsdiagramme für die folgenden Komponenten:

- den Türmechanismus für die Fahrkabine inklusive der Berücksichtigung von Offenhalten, Schnellschließen und den Timer.
- die zentrale Einheit für die Anforderungsbearbeitung (Scheduling der Anfragen)

## Aufgabe 6: Sequence Diagram

Entwickeln Sie ein Sequenzdiagramm für die Fahrt des Fahrstuhls analog zu dem Activity-Diagramm in Aufgabe 2. Dabei soll der externe Auslöser die Anforderung eines Fahrstuhls aus dem Erdgeschoß sein und verschiedene Positionen der Fahrkabinen sind zu berücksichtigen.

Welche Klassen benötigen Sie für diese Anwendung? Welche Klassenattribute wollen Sie verwenden? Das Sequenzdiagramm sollte auch eine grobe Sicht auf die Kontrollstrukturen der jeweiligen Methoden enthalten.

Generieren Sie mit Together automatisch aus dem Sequenzdiagramm eine Implementierung der relevanten Klassen (Skelett). Welche globalen Klassen müssen import werden?

## Aufgabe 7: Testplanung

Skizzieren Sie, wie ein mögliches Konzept für den Test des Systems und seine Komponenten aussehen würde. Orientieren Sie sich dabei an folgenden Punkten:

- Welche Funktionen würden Sie auf der Unit-Ebene testen, welche auf Gesamtsystemebene? Geben Sie jeweils 2 Beispiele an.
- Geben Sie mindestens ein Beispiel für einen Robustheitstest an (d.h. für einen Test der zeigt, dass Fehlbedienungen nicht zu Fehlern führen).

## **Aufgabe 8: Metriken**

Welche der von Together zur Verfügung gestellten Metriken würden Sie in diesem Projekt benutzen, um die Wartbarkeit der entwickelten Software abzusichern. Begründen Sie Ihre Auswahl und verwenden Sie dabei den Goal-Question-Metrics Ansatz; d.h. leiten Sie aus dem Qualitätsziel zunächst Fragen an die Software ab, die dabei helfen, festzustellen, inwieweit das Ziel erreicht ist. Geben Sie dann die Daten an, die diese Frage beantworten könnten.