

Vorgehensweise und Entscheidungsfindung

Zunächst habe ich mir überlegt, welche Klassen ich brauchen kann. Es wird zunächst eine Klasse für den Fahrstuhl und eine für Fahrstuhlschächte gebraucht. Weiterhin soll es eine Klasse für Eingaben und Ausgaben geben und eine Klasse, die den Fahrstuhl letztlich steuert. Die Klasse zum Start des Fahrstuhls könnte man in die Steuerklasse mit einbauen, ich habe mich aber erstmal für eine eigene Klasse entschieden.

class Elevator

In der Fahrstuhlklasse (Elevator) gibt es das Attribut „floors“, für die Anzahl an Stockwerken. Dieses Attribut ist zusätzlich „final“, da es während des Programmstarts einmal mit einer Anzahl an Stockwerken initialisiert wird und danach nicht verändert werden soll.

Das Attribut „actualFloor“ speichert das derzeitige Stockwerk des Fahrstuhls.

Beim Erzeugen eines Fahrstuhl Objekts wird der parametrisierte Konstruktor aufgerufen und die Anzahl an gewünschten Stockwerken übergeben und das „floors“ mit diesem Wert initialisiert. Das Attribut „actualFloor“ wird nicht gesondert initialisiert, da der Fahrstuhl seinen Betrieb ja im Erdgeschoss starten soll. Der default-Wert für ein Attribut vom Typ int ist „0“. Die „0“ entspricht in diesem Programm dem Erdgeschoss.

Mit der Methode „floorUpDown“ werden die einzelnen Stockwerke durchlaufen/durchfahren. Der erste if-Block dient dazu, dass wenn das Zielstockwerk gleich dem aktuellen Stockwerk ist, soll der Fahrstuhl nicht fahren (wie auch...) sondern kund tun, dass er bereits im Zielstockwerk ist. Wird ein anderes Stockwerk gewählt, wird geprüft, ob das aktuelle Stockwerk unterhalb des Gewünschten liegt. Falls dem so ist, fährt der Fahrstuhl aufwärts. Andernfalls (else) natürlich abwärts.

Die Methoden „openDoors“ und „closeDoors“ simulieren das Öffnen und Schließen der Fahrstuhltüren.

Die Methoden „getActualFloor“ und „getFloors“ geben das Stockwerk aus, in dem sich der Fahrstuhl derzeit befindet bzw. wie viele Stockwerke zu Beginn eingelesen wurden.

class ElevatorDuct

Die Klasse „ElevatorDuct“ enthält als Attribut ein Objekt der Klasse Elevator (elevator).

Über den parametrisierten Konstruktor wird ein Elevator-Objekt eingelesen.

Über die Methode „getElevator“ kann das eingelesene Objekt wieder zurückgeliefert werden.

class ElevatorApp

In der Klasse „ElevatorApp“ befindet sich die eigentliche Steuerung des Fahrstuhls.

Die Klasse hat als Attribute ein Array vom Typ ElevatorDuct mit der Arraylänge von 1.

Derzeit ist also nur Platz für einen Fahrstuhl im Schacht.

Weiterhin gibt das Attribut „eleIO“ vom Typ „ElevatorIO“

In der „Start“-Methode wird das Objekt „eleIO“ initialisiert, damit wird auf die Methoden der Klasse „ElevatorIO“ zugreifen können. Als nächstes wird unser neuer Fahrstuhl in dem Array „elevatorDucts“ erzeugt und direkt im Anschluss die Anzahl an Stockwerken eingelesen.

Danach wird nochmals ausgegeben, wo sich der Fahrstuhl befindet, beim Start im Erdgeschoss.

Danach wird die Methode „commandLoop“ aufgerufen, um den Dauerbetrieb zu starten.

Die Methode beginnt mit einer Endlosschleife, über den „Switch“ kann man die gewünschte Aktion wählen. Man kann also so lange Fahrstuhl fahren, bis das Programm explizit über die Eingabe des Buchstaben „b“ beendet wird.

Als Eingaben werden nur die Buchstaben „b“, „e“ und „r“ zugelassen.

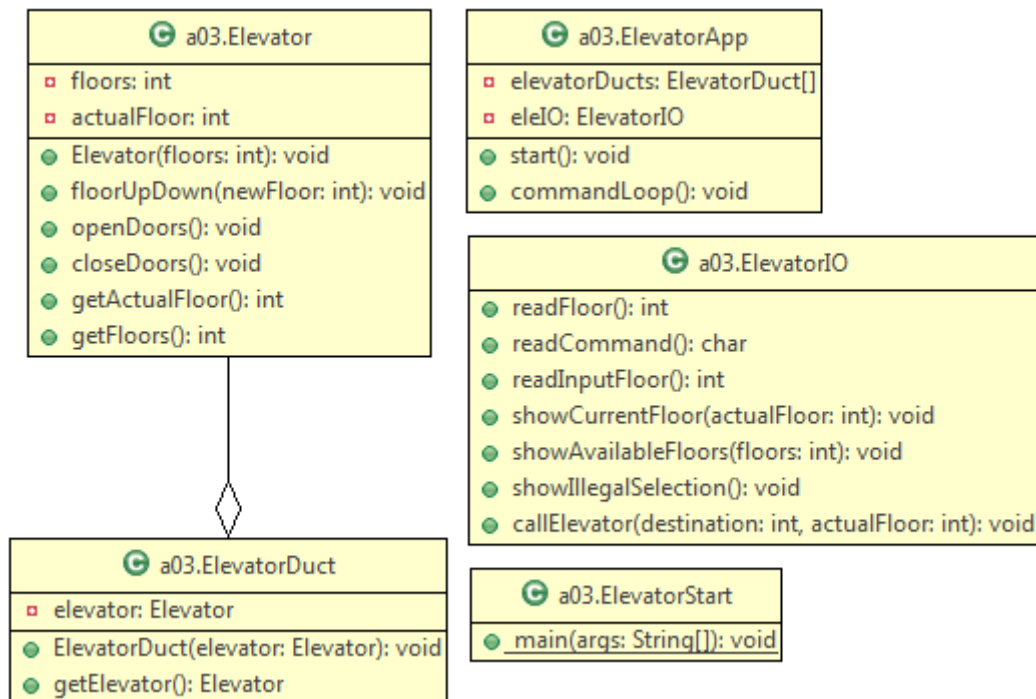
class ElevatorIO

In der Klasse „ElevatorIO“ befinden sich Methoden zum Einlesen von Werten und Ausgeben von Informationen. Die Methoden sind recht einfach gehalten und von der Klasse „Elevator“ entkoppelt, so dass die Klasse „ElevatorIO“ nicht an einen einzigen Fahrstuhl gebunden ist, sondern von beliebigen Fahrstühlen verwendet werden kann.

class ElevatorStart (main-Methode)

Die Klasse „ElevatorStart“ erzeugt in der main-Funktion nur ein Objekt der Klasse „ElevatorApp“ und startet den Fahrstuhl im Anschluss mit der „start“-Methode.

Die main-Methode könnte man auch in die Klasse „ElevatorApp“ integrieren, ich habe mich aber aus Übersichtsgründen dagegen entschieden.



Quellenangaben

Thread Klasse

Sun Microsystems, Inc.; Class Thread

URL: <http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Thread.html>

(abgerufen am 12.04.2010)