
Software Engineering

<http://proglang.informatik.uni-freiburg.de/teaching/swt/2009/>

Aufgabenblatt 6

Aufgabe 1: Klassendiagramm (10 Punkte)

Es soll eine Autovermietung mit folgenden Anforderungen modelliert werden.

- 1 Von jedem PKW wird das Kennzeichen und das Baujahr gespeichert.
- 2 Den einzelnen PKWs werden Typen zugeordnet. Jeder Typ hat eine Bezeichnung und einen Tagesmietpreis. Typen werden auch gespeichert, wenn keine entsprechenden PKWs vorhanden sind. PKWs können nur erfasst werden, wenn der entsprechende Typ auch existiert.
- 3 Zu jedem Auto, das vermietet wird, gibt es einen Entleiher, der gleichzeitig auch Fahrer ist. Darüber hinaus können bis zu drei Zusatzfahrer eingetragen werden. Von den Fahrern (Entleiher und Zusatzfahrer) werden die Namen, Adressen und das Geburtsdatum gespeichert.
- 4 Eine Person darf maximal 5 Autos gleichzeitig (als Entleiher) mieten.
- 5 Zur Vermietung sind keine weiteren Daten zu speichern.
- 6 PKW und Fahrer sollen in dem System gespeichert bleiben, sobald sie einmal erfasst wurden, auch wenn sie aktuell an keiner Vermietung beteiligt sind.

Erstellen Sie anhand dieser Anforderungen systematisch ein vollständiges UML Klassendiagramm.

Aufgabe 2: Statechart Diagramm (10 Punkte)

Die NASA hat Sie als Chefindenieur für die nächste Marsmission angeheuert. Ihre Aufgabe ist es das Kontrollsystem eines autonomen Roboters zu realisieren, der zum Erforschen der Marsoberfläche dienen soll. Nach der Landung wartet der Roboter zuerst auf das Aktivierungssignal aus Houston. Der Roboter soll drei verschiedenen Aufgaben auf dem Planeten Mars durchführen können. Er beginnt damit sobald er das Aktivierungssignal empfangen hat.

Einerseits soll er jederzeit, auf Anforderung von der Erde, Bilder von seiner Umgebung machen und diese zum NASA Hauptquartier senden. Unabhängig davon soll er wiederholt zunächst eine Bodenprobe nehmen und analysieren und danach die Sonneneinstrahlung messen. Jedes Mal wenn im Roboter eine dieser Aufgaben abgeschlossen ist, sendet er auch diese Daten zurück zur Erde.

Falls dem Roboter nur wenig Energie bleibt, unterbricht er seine Aufgaben eine Zeit lang um mithilfe der Solarzellen seine Batterien wieder aufzuladen. Danach setzt er seine Arbeit dort fort, wo er sie unterbrochen hat. Falls der Roboter das Signal zur Deaktivierung empfängt so kehrt er zu der Position zurück, wo er gelandet ist und wartet darauf abgeholt zu werden.

Modellieren Sie das gewünschte Verhalten des Roboters durch ein UML Statechart Diagramm; machen dabei Sie von den Strukturierungsmöglichkeiten wie “composite states” und “concurrent states”, sowie History-Pseudostates dort Gebrauch, wo es angemessen ist.