

# Praktikum II

## Objektorientierte Systeme

Prof. Dr. Nikolaus Wulff

Zum 22. – 29. April 2025

### 1 Spagetti Code oder Design Muster?

Eine der Herausforderungen im Projekts ist, dass der Robot sowohl autonom als auch vom Benutzer gesteuert fahren soll. Für Techniker: Sie benötigen eine **Regelung** ( $\equiv$  autonomes Fahren auf der Line) und eine **Steuerung** ( $\equiv$  Fahren per Fernbedienung) für den Robot - und beide sollen irgendwie sinnvoll miteinander agieren ohne sich gegenseitig zu behindern. Das verlangt förmlich nach einem guten Designmuster.

Da wir in der Vorlesung noch nicht so weit sind hier einige Tips: Der Einsatz des **Strategie** Musters bietet sich an. Eine Strategie ist das Fahren per Steuerung vom Benutzer, eine andere Strategie das autonome Fahren. Im ersten Praktikum haben einige Gruppen bereits einen „Zick-Zack“-Algorithmus zum Fahren auf der der Linie entwickelt - sicherlich noch nicht der ideale Regler. Sinnvoll ist es den besser geeigneten PID Regler einzusetzen. Das Designziel ist es den jeweiligen Algorithmus (ZickZack, PID, UserControl, etc.) in einer eigenen Strategie(klasse) zu kapseln - die allerdings zwingend für das Strategie Muster alle dieselbe Schnittstelle haben müssen - und diesen nicht direkt in dem Lego-Robot als „Spagetti-Code“ zu hinterlegen. So kann die jeweilige Strategie ausgetauscht werden. Noch modularer und flexibler wird das Design, wenn vom System auf Grund eines Ereignis die jeweilige Strategie ausgewählt wird, in dem auf ein Ereignis z.B. durch einen Zustands- und Verhaltenwechsel reagiert wird. D.h. Algorithmen können zur Laufzeit fast beliebig ausgetauscht werden. Ein großer Vorteil des Strategie Musters. Diese Überlegung führt gleich zu weiteren möglichen Design Mustern: **Zustand** mitsamt einem Zustandsautomaten und auslösenden Ereignissen, die z.B. mittels des **Beobachter** Musters detektiert werden können.

Mit Hilfe einer Ereignissteuerung lassen sich die Sensoren des Lego Robots einheitlich und modular auslesen. Gleichzeitig kann/muss diese auch auf Benutzereingaben über die Bluetooth Schnittstelle reagieren. Hierfür kann nutzbringend das **Kommando** Muster eingesetzt werden, um Befehle per Bluetooth zu übertragen.

Registrierte Beobachter können auf Ereignisse reagieren, um z.B. Statusübergänge und ein angemessenes Verhalten einzuleiten. Hierzu ist es notwendig eine Ereignis- und eine Verhaltenssteuerung zu modellieren. Diese sollen lose gekoppelt sein. Ereignis, Verhalten und Status sind unterschiedliche Konzepte, die sinnvoll miteinander agieren und entsprechend modelliert werden.

## Aufgabe

- Erstellen Sie ein UML Modell, das die Ereignis- und Verhaltenssteuerung und mögliche Strategien beinhaltet. Jede Gruppe soll sich zur Vorbereitung auf das Praktikum ein UML Modell überlegt haben **bevor** im Praktikum heldenhaft codiert wird.
- Sehen Sie eine Listener Schnittstelle vor, um die Verhaltenssteuerung und/oder die Statusmaschine als Beobachter registrieren zu lassen.
- Implementieren Sie die Ereignissteuerung und Verhaltenssteuerung für das Verhalten „*fahren auf der Linie*“ oder „*Benutzer steuert per BT*“.
- Das Fahren auf der Linie wird - wie in der Vorlesung besprochen - mit einem PID Regler realisiert für erste Entwürfe kann auch eine ZickZack Strategie hinterlegt werden.

Sobald diese Verhalten grundlegend funktionieren, lassen sich weitere mögliche Verhaltensmuster als zusätzliche Klassen instanzieren, um das Verhalten zu erweitern. Stichwort hierzu ist das „Strategie Muster“

## Tip

Die Ereignissteuerung lässt sich am einfachsten in Form einer Endlosschleife programmieren, die alle benötigten Sensoren kontinuierlich ausliest und deren Werte weitergibt, ähnlich zu der AWT Event Queue einer graphischen Anwendung. Geschickt programmiert kann auch die Bluetooth Kommunikation mit einem „BTSensor“ erfolgen, so dass eine einheitliche Schnittstelle für alle „Sensoren“ vorliegt, notfalls erreicht durch das **Adapter** Muster. Die Verhaltenssteuerung kann dann zu den Ereignissen passende Verhaltensmuster ansteuern, welche z.B. mit einem entsprechenden Status assoziiert sind. Wie Sie das Zusammenspiel zwischen Ereignis-, Verhaltenssteuerung (Strategie) und Statusmaschine modellieren ist die eigentliche Kernaufgabe dieses Praktikums – eventuell helfen Ihnen die im Text oben benannten Design Muster.