

Lernaufgabe

LE 09 – Entwurf mit Design Patterns

Lernziel

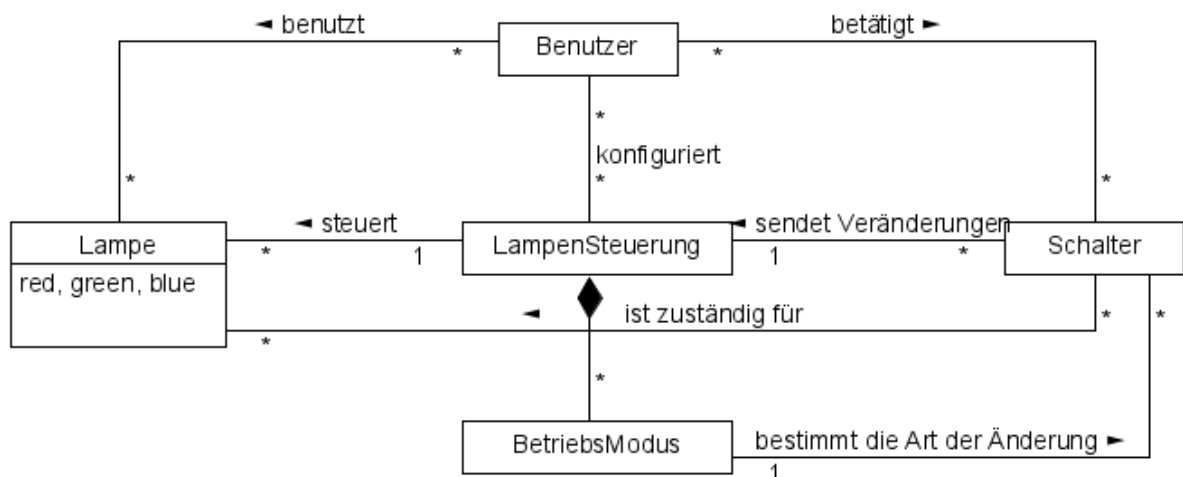
Sie in der Lage, die behandelten Design Patterns an einem konkreten Beispiel anzuwenden.

Einleitung

Diese Lernaufgabe handelt von einer flexibel einsetzbaren Lampensteuerung. Die dafür vorgesehenen Lampen sind bezüglich den Farbkomponenten Rot, Grün und Blau jeweils zwischen 0 und 100% einstellbar. Wie Sie vielleicht wissen, ist jeder Farbton eine Mischung dieser 3 Grundfarben, während Weiss alle 3 Komponenten mit gleicher Stärke hat.

Um das Licht zu steuern, kommen Schalter zum Einsatz, die sowohl gedreht wie auch gedrückt werden können. Allerdings kennen die Schalter ihre aktuelle absolute Position nicht, sie können nur der Lampensteuerung melden, ob Sie gedrückt und in welche Richtung sie gedreht wurden.

Das folgende Domänenmodell soll Ihnen als Einstieg dienen:



Die Lampensteuerung reagiert auf die Schalter-Änderungen und verändert die Lampen, die dem Schalter zugeordnet sind. Die genaue Art der Veränderung hängt vom gewählten Betriebsmodus ab. Ein Betriebsmodus ist fix einem Schalter zugeordnet. Die verfügbaren Lampen und Schalter, die Zuordnung der Lampen zu den Schaltern und der Betriebsmodus für einen Schalter werden beim Start der Lampensteuerung definiert und sind dann konstant. Es ist übrigens möglich, dass eine Lampe von mehr als einem Schalter angesteuert wird.



Um dieses Praktikum überhaupt für jedermann ausführbar zu machen, wurden Software Simulationen für die Lampen und die Schalter entwickelt. Das Ziel schlussendlich sollte aber eine Steuerung sein, die mit richtigen, externen Hardware-Komponenten arbeitet.



Aufgabe 9.1: Analyse des aktuellen Codes

Der aktuelle Stand ist eine reine Simulation, wobei die verschiedenen Klassen direkt miteinander gekoppelt sind. Sie haben dieses Projekt neu übernommen und möchten sich zuerst einen Überblick über den vorhandenen Code verschaffen.

Aufgabe

Analysieren Sie den aktuellen Code auf GitHub https://github.zhaw.ch/SWEN1-LP2019/SWEN1_LE09_Design_Patterns_II_9.1-9.4 und identifizieren Sie Orte, wo Verbesserungen empfehlenswert sind. Sie müssen noch keine Verbesserungen durchführen.

Vorgehen

1. Clonen Sie das oben erwähnte Repository, führen Sie die Anwendung LightControlApp aus und studieren Sie den Code.
2. Zeichnen Sie ein DCD, wobei Sie auf Attribute und Methoden verzichten können.
3. Identifizieren Sie die Orte, wo eine Verbesserung (Refactoring?) empfehlenswert ist.
4. Welche Design Patterns ziehen Sie dazu bei?

Hinweise, Tipps

- Sie sollten alle Konzepte des Domänenmodells in der Software wiederfinden, mit Ausnahme der Konzepts Benutzer und Betriebsmodus. Der Benutzer ist ein Akteur und wird, basierend auf den aktuellen Anforderungen, nicht in der Software abgebildet. Um den Betriebsmodus werden wir uns später kümmern.
- Beachten Sie, dass es ja nicht bei der reinen Simulation bleiben wird sondern dass richtige HW Komponenten zum Einsatz kommen werden. Dennoch kann es Sinn machen, im Sinn eines gemischten Betriebs auch mit den Simulationen weiterzuarbeiten.
- Legen Sie den Schwerpunkt auf Verbesserungen der Kopplungen.
- Sie brauchen die Zugriffe auf Container Objekte (die Objekte, die die 1..* Beziehung realisieren) nicht im Detail zu modellieren. Wenn Sie dies so machen, sollten Sie aber Methoden einfügen, die diese Zugriffe abkapseln oder Sie vermerken über Notizen, wo Sie auf die Container Objekte zugreifen.

Ergebnis

Ein Design-Klassendiagramm der aktuellen Situation mit Bemerkungen über die Verbesserungsmöglichkeiten.

Zeit: 30'

Aufgabe 9.2: Verbesserung des aktuellen Codes

Nachdem wir die kritischen Punkte identifiziert haben, möchten wir diese auch verbessern.

Aufgabe

Verbessern Sie die Kopplungen derart, dass für die Schalter und Lampen später die richtigen Komponenten eingesetzt werden können.

Vorgehen

1. Wählen Sie geeignete Design Patterns und Indirection-Varianten aus, um die Schalter und Lampen und Ihre Kopplungen zu anderen Klassen zu verbessern.
2. Visualisieren Sie Ihre Überlegungen mit einem UML Klassen- und Interaktionsdiagramm.

Hinweise, Tipps

- Die richtigen Schalter und Lampen haben ein eigenes Mikroprozessorsystem und werden zum Beispiel über WLAN oder Bluetooth angesteuert. Was genau zum Einsatz kommt, ist im Moment nicht relevant.
- Die Mikroprozessorsysteme der Schalter und Lampen sind so einfach, dass nur ein minimales Programm dort Platz findet und Aktualisierungen nur sehr mühsam möglich sind. Daher sollte ihre Verantwortlichkeit so einfach wie möglich sein.
- Im Moment reicht es aus, dass wir nur einen RotarySwitch haben und der steuert alle Lampen an.
- Sie brauchen die Zugriffe auf Container Objekte (die Objekte, die die 1..* Beziehung realisieren) nicht im Detail zu modellieren. Wenn Sie dies so machen, sollten Sie aber Methoden einfügen, die diese Zugriffe abkapseln oder Sie vermerken über Notizen, wo Sie auf die Container Objekte zugreifen.

Ergebnis

Ein aktualisiertes Design-Klassendiagramm und ein Interaktionsdiagramm mit Hinweisen auf verwendete Design Patterns.

Zeit: 30'