

Frage: Mir ist der View-Begriff beim MVC im Zusammenhang mit dem Beispiel mit den Plus- und Minus-Tasten unklar.

Zunächst ist wichtig, dass es sich bei MVC um ein Konzept handelt, eine oder mehrere Klassen übernehmen die Aufgaben des Views des Controllers und des Modells. Bei dem graphischen Beispiel ist es irritierend, dass es eigentlich zwei Modelle gibt. Einmal der verwaltete Wert im XModel-Objekt und ein Modell für die graphische Oberfläche (z. B. Buttonfarbe, Buttonposition, Buttonzustand, Buttongröße, ...). Der Button hat damit seine eigene View, die diesen Button auf den Bildschirm malt. Das ist aber bei der Aufgabe weniger vom Interesse. Hier ist das GUI etwas vereinfacht die View zum XModel. Views stellen Modelle dar, nehmen Befehle an, können eventuell eine (Teil-)validierungen vornehmen und den Controller benachrichtigen. Eventuell wird die Vorstellung einfacher, wenn das GUI durch eine Konsolen-Ein-und-Ausgabe ersetzt wird. Dann stellt der View das Model dar und ermöglicht den Zugang zu Controller-Aktionen. Wenn Sie eine enge Verwandtschaft zur Boundary bei Boundary-Control-Entity spüren, passt das, der View zeigt zusätzlich etwas in irgendeiner Form an und wird meist bei GUIs genutzt. Als Hintergrund vielleicht interessant: <https://stackoverflow.com/questions/32912341/entity-control-boundary-ecb-vs-model-view-controller-mvc>

Frage: Welche Informationen stelle ich im Sequenzdiagramm dar?

Antwort: Die für die Beschreibung relevanten. Genauer, stehen in der Kopfzeile des Diagramms alle wichtigen Objekte, die nicht im Laufe des Diagramms erzeugt werden. Woher die Objekte kommen ist damit für dieses Diagramm unwichtig. Dann zeichnen Sie alle Methoden detailliert ein, die zum Verständnis der dokumentierten Funktionalität wichtig sind. Gleiches gilt für Rückgabepfeile, auf die verzichtet werden kann, wenn es void-Methoden sind oder die Information irrelevant sind. Es muss aber nicht darauf verzichtet werden, wenn der bekannte rote Faden durch das Diagramm erkennbar sein soll. Generell darf nichts dazu erfunden werden, wenn ein detaillierter Ablauf $m_1; m_2; m_n$ gegeben ist, sind alle Folgen wie $m_2; m_4; m_5; m_8$ generell denkbar, solange die Reihenfolge eingehalten wird. Generell ist es natürlich auch möglich UML-Kommentarkästen in das Diagramm zu setzen.